CRWR Online Report 97-6

Water Quality Master Planning for Austin

by

Christine Michele Dartiguenave, Ingénieur ECLille

Graduate Research Assistant

and

David R. Maidment

Principal Investigator

December 1997

CENTER FOR RESEARCH IN WATER RESOURCES

Bureau of Engineering Research • The University of Texas at Austin
J.J. Pickle Research Campus • Austin, TX 78712-4497

This document is available online via World Wide Web at
http://www.ce.utexas.edu/centers/crwr/reports/online.html

Acknowledgements

I would like to thank my advisor, Dr. David Maidment, for his support through this project. I am also grateful to Dr. Francisco Olivera and Dr. Michael Barrett for their constant assistance throughout this research.

I would like to thank the City of Austin, which funded this project, and more particularly Ed Peacock, Pat Hartigan, Leila Gosselink and Ellen Wadworth. I am especially grateful to Ellen for her comments and for her friendship.

I have much appreciated the help and support I received from the members of Dr. Maidment's GIS Research team: Kwabena Asante, Ferdinand Hellweger, Connie Hinojos, Cindy How, Ann Quenzer, Seann Reed and Maggie Ye Ruan.

Last but not least, I would like to thank my family, who is always there for me.

December 1997

Abstract

Water Quality Master Planning for Austin

Christine Michele Dartiguenave, M.S.E.

The University of Texas at Austin, 1997

Supervisor: David R. Maidment

The goal of this research is the creation of a non-point source pollution water quality model using a Geographic Information System. The area chosen for the study is the City of Austin, which partly overlays the recharge zone of the Edwards Aquifer. A model based on raster data that takes into account the presence of the recharge zone was created both in ArcView and in Arc/Info for mean annual flows and pollutant loadings. The model is able to perform the following tasks: 1) compute current pollutant loadings for TSS, BOD, COD, TOC, DP, TP, NH_3, TKN, NO_3, TN, Cu, Pb and Zn, 2) compute future loadings for the year 2040 for the same constituents, 3) model the effect of located and regional Best Management Practices. The model was designed so that it could deal with different sets of input parameters and locations.

Table of Contents

Table of Contents (cont.)

Table of Contents (cont.)

Table of Contents (cont.)

List of Tables

List of Figures

List of Figures (cont.)

List of Figures (cont.)

List of Figures (cont.)

List of Figures (cont.)

List of Procedures

Chapter 1: Introduction

1.1 BACKGROUND

Non-point source pollution is a leading cause of water quality problems. For a long time it was considered negligible in comparison with point-source pollution for which extensive legislation has been developed. The importance of non-point source pollution in water quality issues was eventually acknowledged in the 1987 Amendment to the Clean Water Act, which created the section 319 Non-Point Source Management Program. The Environmental Protection Agency defines non-point source pollution as water pollution being caused by rainfall or snowmelt moving over and through the ground. The runoff carries natural and human-made pollutants deposited on the land surface to receiving waters.

Beginning in the early 1970's, the City of Austin has implemented a storm runoff-monitoring program. The data gathered over many years of operation since then constitute an important and unique source of information on storm water quality in the City. The City has also required the installation of Best Management Practices (BMPs), such as ponds or sand filters, to reduce pollutant loading.

Because the quantity of pollutants deposited is directly related to the level of development, the growth of Austin will increase the impact of non-point source pollution on urban creeks. To find solutions to the non-point source pollution problem, the City desires to determine the actual and projected non-point source loadings at a set of eighty Environmental Integrity Index sites. These sites are locations on streams which are periodically sampled and assessed using an Environmental Integrity Index comprised of physical, chemical and biological measures of environmental quality (City of Austin Drainage Utility Department, 1997).

Austin's storm water monitoring program does not provide information on pollution loading for the entire City. The research presented here includes extrapolating observed non-point source loading at monitoring sites to the eighty index sites and calculations for both current and future conditions, where future conditions correspond to the land use predicted for the year 2040. Current (1990) land use conditions estimated by the City were used then assumptions were developed for various urbanization scenarios for the next 50 years to arrive at year 2040 conditions.

Figure 1.1 defines the location of the study area in Texas and Figure 1.2 shows the location of the sample sites used in this study, which are the United States Geological Survey (USGS) stations, and the Environmental Integrity Index (EII) sites. The study region in Figure 1.2 is defined by the watersheds under study in the project and is described in more detail in Chapter 2.

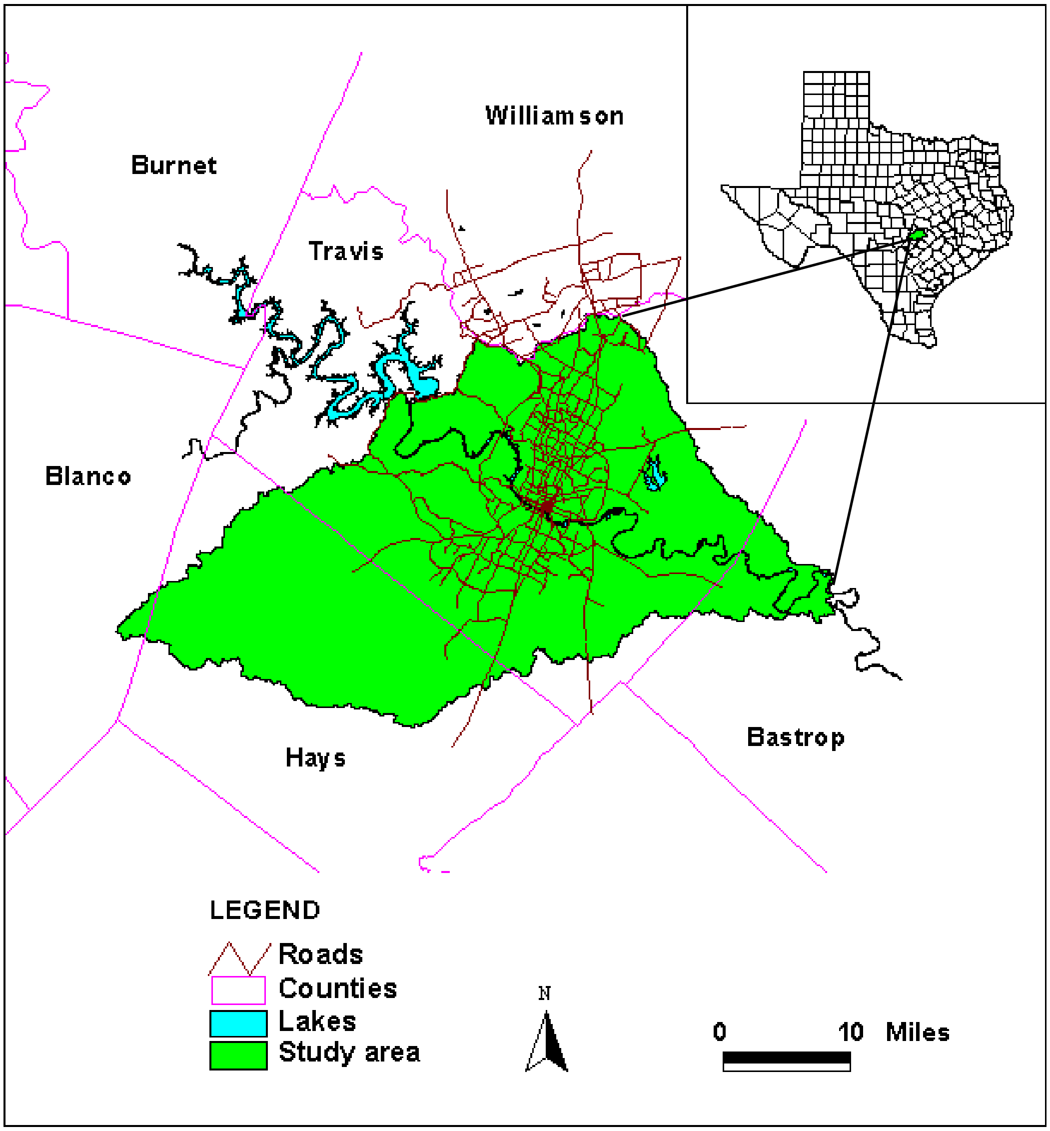


Figure 1.1: Study area

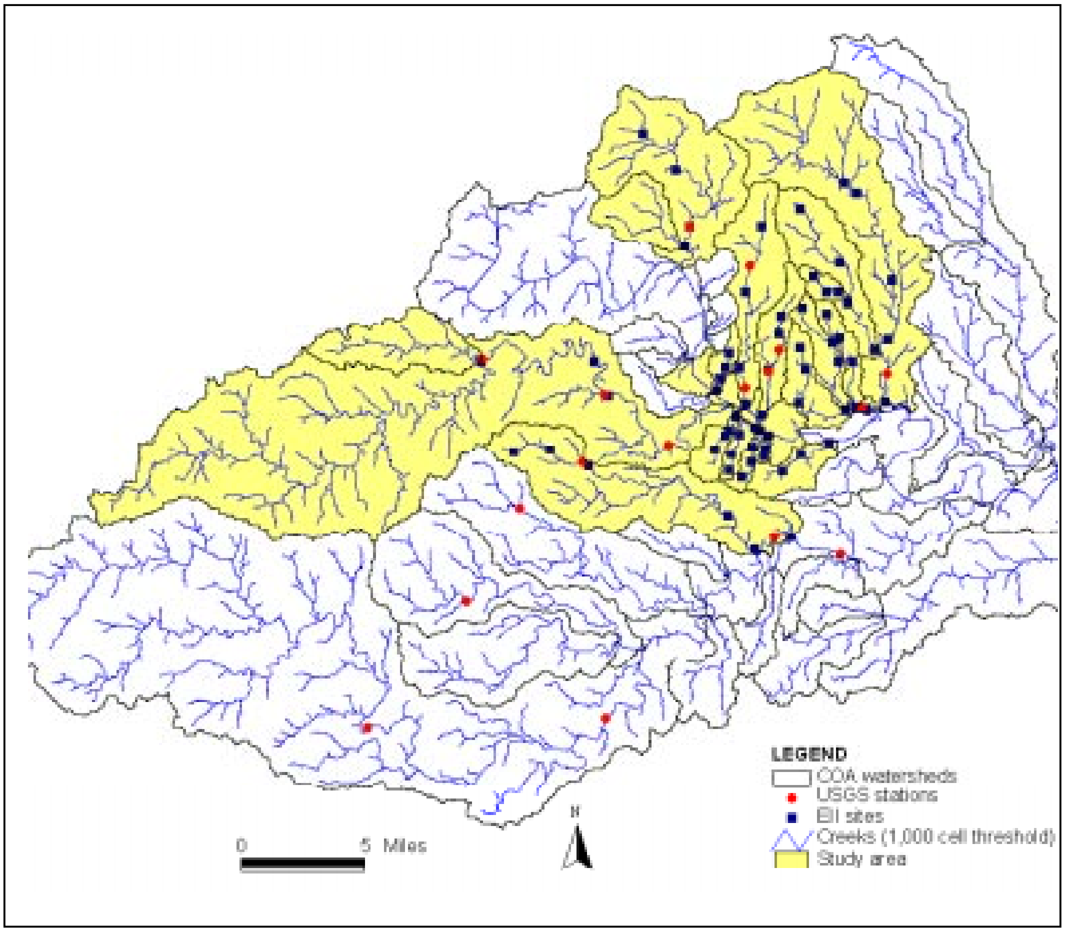


Figure 1.2: Location of study sites in drainage area

1.2 METHODOLOGY

Previous studies conducted at the Center for Research in Water Resources have shown GIS to be a valuable tool in the analysis of non-point source pollution problems. Pawel Mizgalewicz modeled agrochemical transport in the Iowa-Cedar basin (Mizgalewicz and Maidment, 1996) and Bill Saunders did an assessment of non-point source pollution in the San Antonio-Nueces basin (Saunders and Maidment, 1996). Both studies used Geographic Information Systems (GIS) to characterize the land surface (land use, impervious cover…). GIS is a computer system capable of storing and using data describing locations at the surface of the earth. Information can be associated with an object on a map and spatial operations can be conducted on that object. Operations such as union, intersection or selection according to specific criteria can be implemented. The GIS software programs used in these projects are Arc/Info and ArcView, developed by the Environmental Systems Research Institute (ESRI).

The input parameters necessary to determine non-point source pollution loading include precipitation and land use. Relationships derived from the observed data are used to link land use to impervious cover, and impervious cover to runoff coefficients and Event Mean Concentrations (EMCs). Figure 1.3 illustrates this concept.

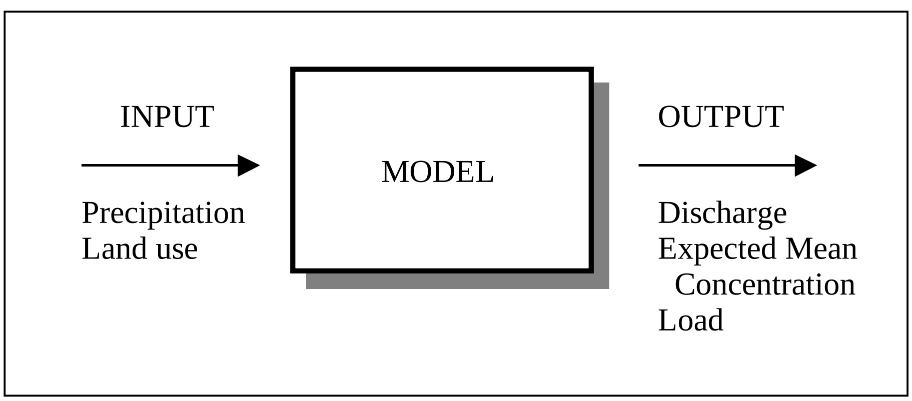


Figure 1.3: Non-point source pollution model

GIS converts a vector representation (coverage) of the impervious cover for a given area to a raster representation. By applying the necessary relationships, a runoff coefficient grid and an EMC grid can be derived. The load produced by each cell can then be determined by multiplying these grids with a precipitation grid:

load = runoff coefficient * EMC * precipitation volume

However, the path followed by the water through the landscape is needed to determine the total loading contribution at any cell. The water flows in the direction of steepest descent, which is determined by the topography of the area. A Digital Elevation Model (DEM), which consists of a sampled array of elevations at regularly spaced intervals (Figure 1.4), is used to determine the path of the water as described in more detail in Chapter 3.

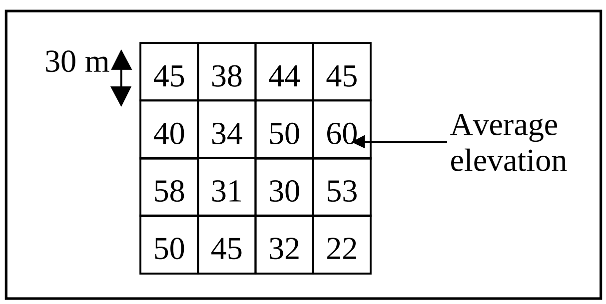


Figure 1.4: Digital Elevation Model

There are three main differences between this and the two previous CRWR studies cited earlier. The first is the size of the study area. The fact that the Austin area is considerably smaller means that a smaller cell size for the grid was needed to get a good representation of the topography. This greater resolution was possible because of the availability of topographic data with a 30-meter cell size. The second difference is the inclusion of base flow. It was assumed in the Austin study that the path of the base flow was identical to the path of the direct runoff but that they have different pollutant concentrations. The third difference is that this study assesses the effects of Best Management Practices on loadings.

1.3 ASSUMPTIONS

The general methodology presented above implies a few assumptions and choices, concerning the time scale, the surface representation and the processes affecting the fate of a constituent.

- **Time scale**

All calculations are done on a mean annual basis. Non-point source pollutants are carried by storm water runoff. Since runoff coefficients are directly related to land use and since EMC values are either related to land use or are constant, the pollutant load is directly proportional to the amount of rainfall. In the Austin area, precipitation varies widely from year to year. Hence the average annual loading value computed over several years indicates the expected values of the annual loadings which occur but does measure the variability of that loading. The objective of the study is not to define a very accurate value for the loading but to rank the different watersheds to determine the areas with greater problems. Moreover, the advantage of the annual scale is that the time lag between rainfall and flow does not have to be considered. The analysis is spatially intensive but time averaged.

Assuming that the parameters used are not time dependent (i.e. the EMCs are constant), monthly loading values can be generated by weighting the total load by the percentage of rainfall for each month. Further studies are being conducted to test this assumption.

- **Non-point source versus point source discharge**

The model assumes that the storm water follows the natural topography to reach the receiving water. However, in reality, the water can be captured at an inlet and can discharge through a storm drain. Storm sewers can modify the location where the discharge and water quality impacts occur, but the total loading in the creek remains the same if the drains discharge to the same watershed. A major problem in modeling occurs where the sewers collect the water from one watershed and discharge to another. As no detailed description of the storm sewers system exists for Austin, they have not been considered in this study.

- **Conservative constituents**

All the constituents are considered to be conservative: there are no losses due to chemical reactions or biochemical degradations (true for total dissolved solids (TSS) and dissolved metals). The short travel time to the Environmental Integrity Index sites justifies ignoring decay.

Chapter 2: Creation of a base map

This model requires a spatial representation in a Geographic Information System of the study area. The first step in constructing this representation consists in locating the area of interest and representing it in the GIS. It is then essential to correctly locate the points representing the monitoring stations and the EII sites.

2.1 AREA OF STUDY

2.1.1 Location

Austin is located in south central Texas at approximately 98ºW longitude and 30ºN latitude (Figure 1.1). The study area is the drainage area of the Colorado River that ranges from the outlet of Lake Travis at Mansfield Dam to the junction of Wilbarger Creek with the Colorado River, comprising approximately 2,300 km^2 and forty five watersheds (Table 2.1 and Figure 2.1).

Table 2.1: Watersheds within the DEM study area

Barton	Country Club	Harris Branch	Onion	Waller
Bear	Decker	Huck's Slough	Rinard	Walnut
Bee	Dry	Johnson	Shoal	West Bouldin
Blunn	Dry North	Lake Austin	Slaughter	West Bull
Boggy	Eanes	Little Barton	South Boggy	Williamson
Bull	East Bouldin	Little Bear	South Fork	
Buttermilk	Elm	Little Bee	Tannehill	
Carson	Fort Branch	Little Walnut	Taylor Slough No	
Colorado	Gilleland	Marble	Taylor Slough So	
Cottonmouth	Harper's Branch	North Fork	Town Lake	

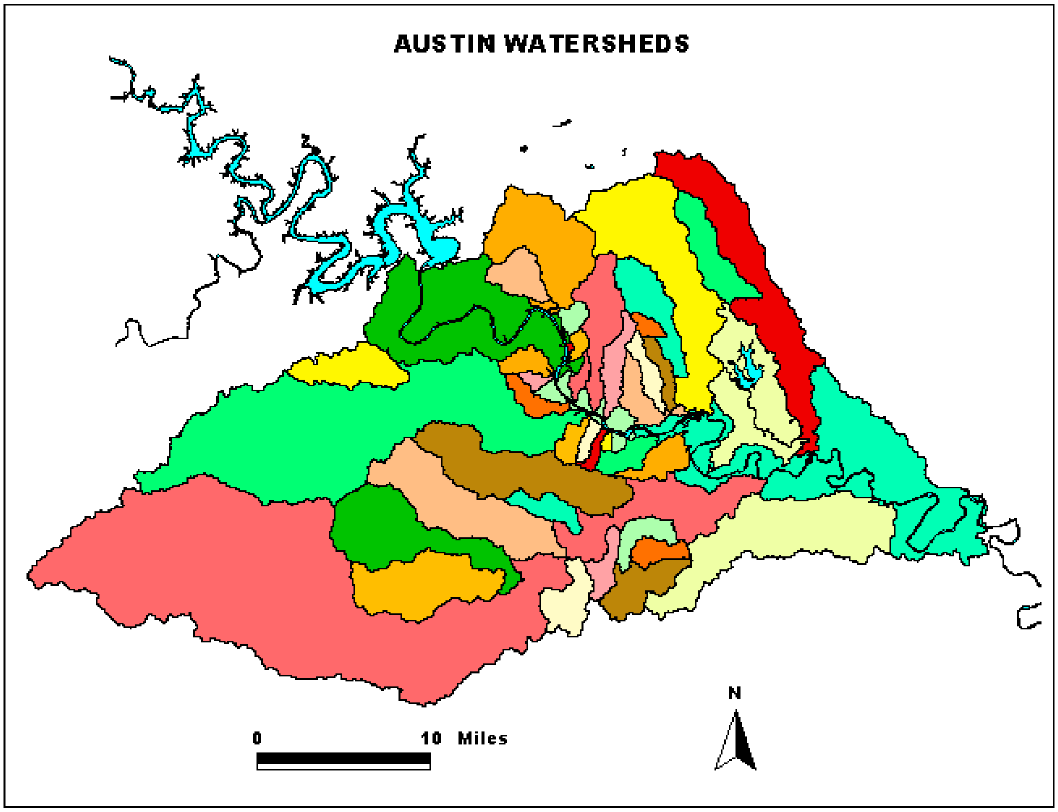


Figure 2.1: Austin watersheds

This area covers part of Travis County, where Austin is located, as well as part of Bastrop, Hays and Blanco counties (Figure 1.1). The county boundary between Travis and Williamson counties coincides with the boundary of the drainage area.

It is also important to note the location of the recharge and contributing zone of the Edwards Aquifer (Figure 2.2). There is a difference between the northern recharge zone (north of the Colorado River) and the southern recharge zone (south of the Colorado River). While extensive studies have been conducted in the southern recharge zone, due to the presence of Barton Springs, little has been done on the northern recharge zone.

Hence this study will only estimate the effect of the southern recharge where more stringent rules have been established.

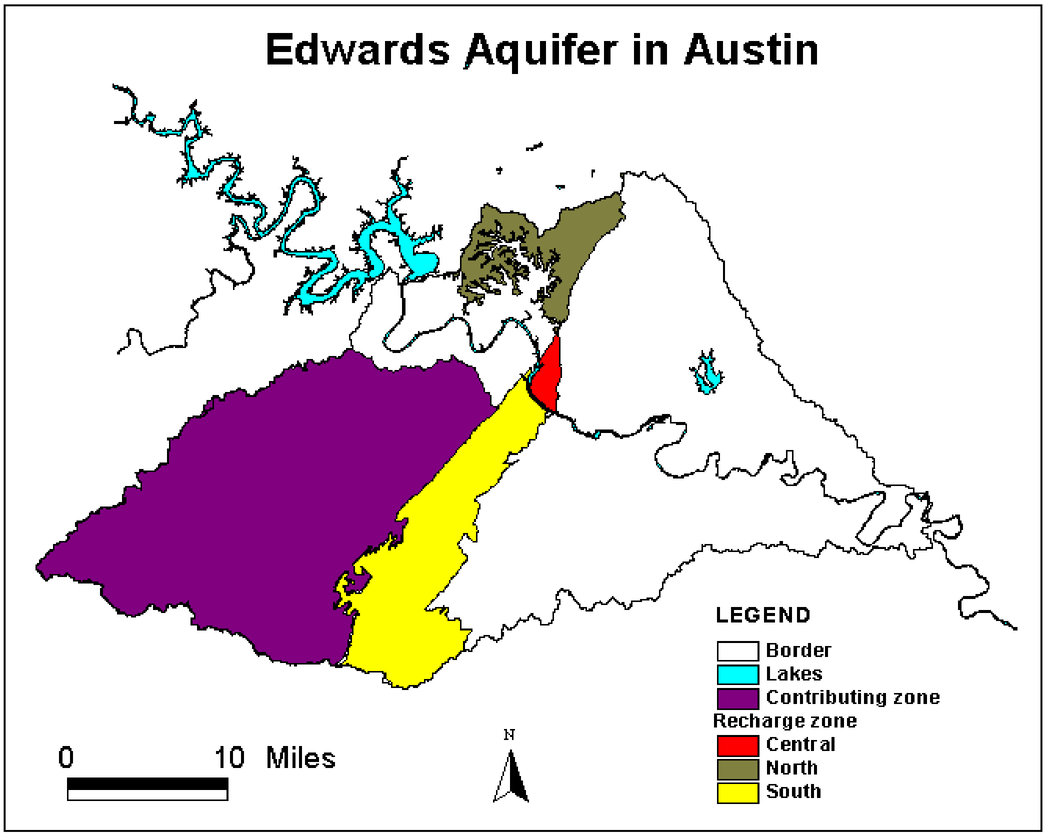


Figure 2.2: Edwards Aquifer in Austin

The objective of the City of Austin is to estimate pollutant loads in 18 watersheds (including 2 subwatersheds) within the study area. These watersheds, shown in Figure 2.3, are (Table 2.2):

Table 2.2: Watersheds under study

Barton	Country Club	Little Barton	Waller
Blunn	East Bouldin	Little Walnut	Walnut
Boggy	Fort Branch	Shoal	West Bouldin
Bull	Harper's Branch	Tannehill	West Bull
Buttermilk	Johnson	Town Lake	Williamson

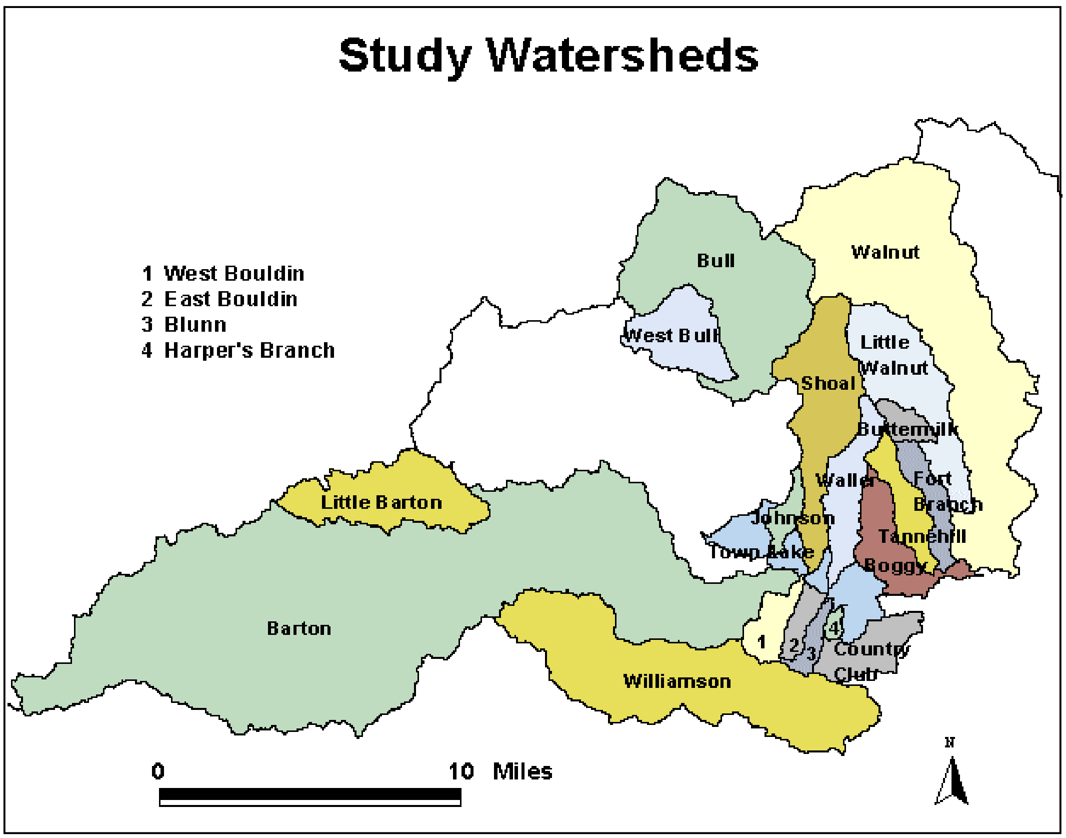


Figure 2.3: Study Watersheds

Two subwatersheds drain to watersheds in the study area and are considered as well. The load and discharge from these watersheds must be taken into account in the total load.

These two subwatersheds are:

- West Bull, included in Bull
- Little Barton, included in Barton

During the study, Eanes watershed was added to the list of watersheds. Since the computations have been conducted for the whole area defined in Figure 2.1, adding this watershed did not required any additional study.

The size of the region does not affect the methodology, so it is convenient to define the whole area as a watershed draining toward the Colorado River. This watershed was delimited in Arc/Info and contains approximately 2.5 million 30-meter cells: its drainage area is approximately 2,300 km^2.

2.1.2 Projection: Texas State Mapping System

For a small scale study such as this, the projection chosen system is the State Plane Mapping System because it is the standard map projection for the City of Austin. In the United States, the State Plane System was developed in the 1930s. It was based on the North American Datum 1927 (NAD27) whose coordinates are based on the foot as the length measure. It was developed in order to provide local reference systems that were tied to a national datum. Small States use a single State Plane zone while larger states such as Texas are divided into several zones. Different projections are used depending on the extent of the States. Lambert Conformal Conic projections are used for rectangular zones with a larger east-west than north-south extent. Transverse Mercator projections are used to define zones with a larger north-south extent.

While the NAD-27 State Plane System has been superseded by the NAD-83 System (based on the meter), maps in NAD-27 coordinates are still in use. At the beginning of the study the coverages provided by the City of Austin were defined with NAD27, so this datum was chosen for the study. Table 2.3 describes the projection file used. There are five State Plane zones in Texas. The zone 5376, covering Central Texas, is used in this study

Table 2.3: Study projection file

projection state
units feet
zone 5376
datum NAD27
parameters
end

During the course of the study, the City of Austin chose to switch to NAD83. As it would have meant redefining almost everything accomplished up to that point, the datum NAD27 was retained for the study. At some point, all the data should be converted to the newer projection. However, the results described in this document will remain valid.

2.2 USGS STATIONS

2.2.1 Locations

The locations of the USGS stations located in the study area (32 stations, Table 2.4) were obtained from the USGS web site related to the Water Data for Texas (http://txwww.cr.usgs.gov/cgi-bin/txnwis). Their geographic coordinates are given in degrees, minutes and seconds.

Table 2.4: USGS stations

USGS #	Name	Latitude	Longitude	Period of record of daily-mean discharge
8154510	Colorado River below Mansfield Dam	30° 23' 30"	97° 54' 28"	10/01/1974-09/30/1990
8154700	Bull Creek at Loop 360	30° 22' 19"	97° 47' 04"	07/18/1978-active
8154900	Lake Austin	30° 18' 53"	97° 47' 10"	Not available
8155200	Barton Creek at SH 71 Near Oak Hill	30° 17' 46"	97° 55' 31"	02/07/1978-10/15/1982 and 01/01/1989-active
8155240	Barton Creek at Lost Creek Blvd	30° 16' 28"	97° 50' 39"	12/28/1988-active
8155260	Barton Creek near Camp Craft Road	30° 16' 12"	97° 49' 43"	09/01/1982-10/11/1988
8155300	Barton Creek at Loop 360	30° 14' 40"	97° 48' 07"	02/01/1977-active
8155550	West Bouldin Creek at Riverside Drive	30° 15' 49"	97° 45' 17"	04/21/1985-04/22/1985, 06/05/1985-06/07/1985
8156700	Shoal Creek at NW. Park	30° 20' 50"	97° 44' 41"	03/28/1975-09/30/1984
8156800	Shoal Creek at West 12th Street	30° 16' 35"	97° 45' 00"	01/08/1983-07/27/1983 and 09/07/1983-active
8157000	Waller Creek at 38th Street	30° 17' 49"	97° 43' 36"	04/01/1955-10/23/1980
8157500	Waller Creek at 23rd Street	30° 17' 08"	97° 44' 01"	01/01/1955-10/23/1980
8158000	Colorado River at Austin	30° 14' 40"	97° 41' 39"	03/01/1898-active station
8158050	Boggy Creek at US Highway 183	30° 15' 47"	97° 40' 20"	03/02/1976-05/25/1976, 01/25/1977-03/09/1977, 06/16/1977-09/30/1986
8158100	Walnut Creek at FM 1325	30° 24' 35"	97° 42' 41"	10/19/1984-10/23/1984, 05/13/1985-05/15/1985, 10/18/1985-10/20/1985, 09/05/1986-09/07/1986
8158200	Walnut Creek at Dessau Road	30° 22' 30"	97° 39' 37"	05/13/1985-05/15/1985, 02/03/1986-02/04/1986, 05/16/1986-05/18/1986
8158300	Ferguson Branch at Springdale Road	30° 19' 53"	97° 39' 12"	Not available
8158380	Little Walnut Creek at Georgian Drive	30° 21' 15"	97° 41' 52"	02/21/1985-02/24/1985, 05/12/1985-05/13/1985, 10/18/185-10/20/1985, 04/29/1986-05/02/1986
8158600	Walnut Creek at Webberville Road	30° 16' 59"	97° 39' 17"	05/27/1966-active
8158820	Bear Creek at FM Rd 1626 near Manchaca	30° 08' 25"	97° 50' 50"	Not available
8158840	Slaughter Creek at FM Rd 1826	30° 12' 32"	97° 54' 11"	01/16/1978-active
8158880	Boggy Creek at Circle S Road	30° 10' 50"	97° 46' 55"	06/05/1985-06/06/1985, 10/14/1985-10/15/1985, 05/15/1986-05/15/1986

Table 2.4 (continued): USGS stations

USGS #	Name	Latitude	Longitude	Period of record of daily-mean discharge
8158920	Williamson Creek at Oak Hill	30° 14' 06"	97° 51' 36"	01/10/1978-03/08/1993
8158922	Williamson Creek at Brush Country Blvd	30° 13' 34"	97° 50' 28"	03/11/1993-active
8158927	Williamson Creek at Brush Country Road	30° 13' 36"	97° 50' 28"	10/01/1984-09/30/1985
8158930	Williamson Creek at Manchaca Road	30° 13' 16"	97° 47' 36"	10/01/1984-09/30/1985
8158970	Williamson Creek at Jimmy Clay Road	30° 11' 21"	97° 43' 56"	09/11/1975-09/30/1986
8159000	Onion Creek at US Highway 183	30° 10' 40"	97° 41' 18"	06/01/1924-02/28/1930, 03/23/1976-active
8158700	Onion Creek near Driftwood	30° 04' 59"	98° 00' 29"	07/01/1979-active
8158800	Onion Creek at Buda	30° 05' 09"	97° 50' 52"	07/01/1979-09/30/1983 and 01/01/1992-active
8158810	Bear Creek below FM Rd 1826 near Driftwood	30° 09' 19"	97° 56' 23"	07/07/1979-active
8158825	Little Bear Creek at FM Road 1626 near Manchaca	30° 07' 31"	97° 51' 43"	Not available

To be used in the GIS system, these data were first converted to decimal degrees according to the equation:

$$\text{Decimal Degrees (DD)} = \text{Degrees} + \frac{\text{Minutes}}{60} + \frac{\text{Seconds}}{3600}$$

The coordinates are stored in a text file (*stat.dat*) shown in Table 2.5, which is used to build the point coverage in Arc/Info.

Table 2.5: USGS stations coordinates in decimal degrees (stat.dat)

```
1 –98.0806 30.4208
2 –97.9078 30.3917
3 –97.7844 30.3719
4 –97.7861 30.3147
5 –97.9253 30.2961
…
end
```

2.2.2 Generate the point coverage

- **Step by step**

The point coverage was built in Arc/Info from the data in a text file (e.g. stat.dat)

Arc: **generate stations**
Generate: **input stat.dat**
Generate: **points**
Generate: **quit**
Arc: **build stations point**
Arc: **addxy stations**

The result is a point coverage of the stations in geographic coordinates, which is then converted to the State Plane projection system used in the study. The input and output parameters of the projection from geographic to state plane are written in a text file (sta_prj) used in the projection (Table 2.6).

Table 2.6: Conversion from geographic to state plane coordinates (sta_prj)

```
Input
Projection geographic
Datum NAD27
Units dd
Parameters
Output
Projection state plane
Zone 5376
Datum NAD27
Units feet
Parameters
End
```

The coverage *stations* in geographic coordinates can now be converted into the coverage *station* in state plane coordinates.

Arc: **project cover stations station sta_prj**

- **Using Arc Macro Language**

Arc Macro Language (AML) is the programming language of Arc/Info. AML enables one to write a program which will automate many tasks. To create a point coverage, an AML called *ptcov.aml* (Procedure 2.1) was created in a text editor.

Procedure 2.1: Create a point coverage in state plane coordinates (ptcov.aml)

```
Generate stat
Input [response'name and path of the input file']
Points
Quit
Build stat points
Addxy stat
Project cover stat [response'name and path of the coverage'] sta_prj
&return
```

This program is run in Arc by using the AML command &run:

Arc: **&run ptcov**

The program will prompt for the name and path of the input file and the new point coverage.

2.2.3 Adjust the coverage

The USGS stations are used to determine the actual flow in the creeks; therefore, it is important that their gage locations are accurately located in the stream network grid representing the delineated creeks (section 3.4). However, this is often not the case and so corrections have to be made with either Arc/Info (ArcTools) or ArcView.

- **Adjustment of gage locations with Arctools**

Arview3 enables one to check if a station is located on a creek by a point by point verification, but it can not be used to modify the position of the stations. Only Arc/Info allows an adjustment of the data by making corrections in the associated grid using Arctools. The coverage of the stations first must be converted to a grid, whose points will be compared with a stream network grid.

Arc: **grid**
Grid: **station_gr = pointgrid (station)**

ArcView is then used as an interface to display these grids to check if the stations are located in the creeks. If it is not true, then the grid must be modified in order for the stations to be at the "right" place. Once the grid with the stations has been manually corrected, it is reconverted into a point coverage.

Grid: **station_sta = gridpoint (station_gr)**

However, the Arctools display environment is not very user friendly, and it is quite time-consuming to swing back and forth between ArcView, and Arctools.

- **Adjustment of gage locations with ArcView**

The process described above is time consuming however. ArcView has its own programming language called Avenue, which enables the user to develop new functions. An avenue program, also called a script, can be used to create the desired function and hence be able to solve the problem in a better computing environment.

The first step is to create a program which quickly and automatically determines if a station is correctly located in the creek. The principle is to use a grid of the streams where the value of the cells containing the creeks is one. All other cells are no data cells (Figure 2.4). The value of the grid at the location of the stations indicates wether they are correctly located in the creeks.

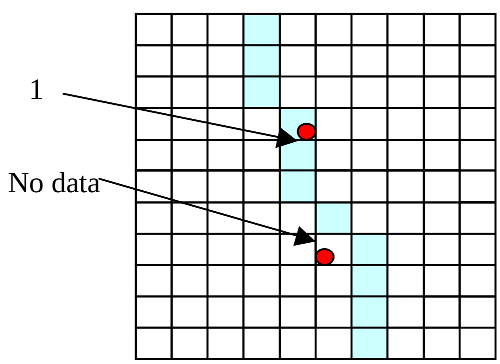
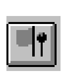
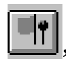


Figure 2.4: Station location

This process is done by using a script called Qual.Pick (Appendix C), customized with the button  which is accessible when the View window is active.

This program allows one to obtain the values of several grids at the locations given by a point coverage. When clicking on , a series of message boxes prompt the user for the grid(s) (Figure 2.5) and the point coverage (Figure 2.6) to use in the program.

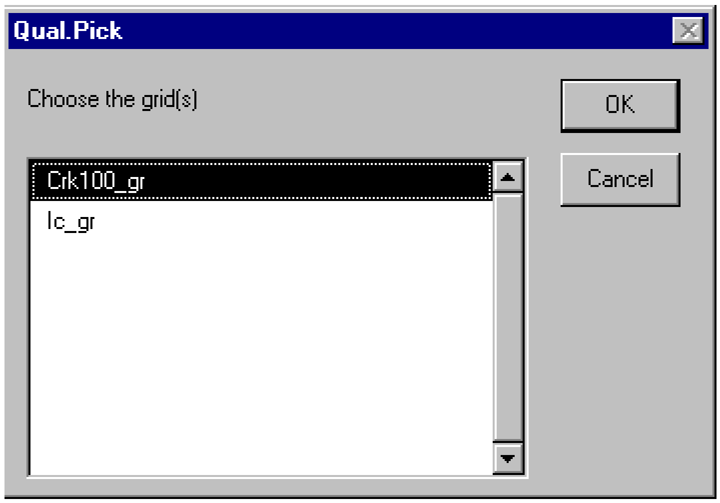


Figure 2.5: Choose the grids (Qual.Pick)

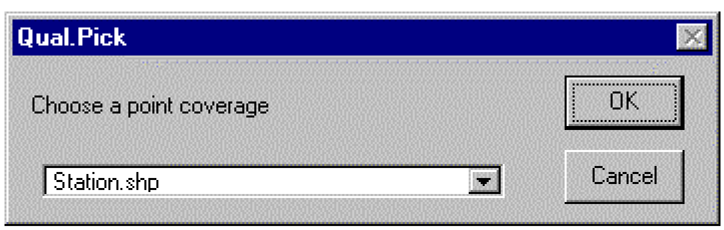


Figure 2.6: Choose a point coverage (Qual.Pick)

Several grids may be selected. However in this case, the only grid to select (crk100_gr) is a creek grid whose cell value is one in the creeks and no data elsewhere, the creeks being defined as the cells with a drainage area bigger than 100 cells. For each point in the point coverage, the program writes the value associated to the grid cell where the point is located to a new field added to the attribute table of the point coverage. By default, the name of the grid the value is taken from is also the name of the field. If the field already exists, a message box prompts the user either to choose a new name or to overwrite the existing field (Figure 2.7).

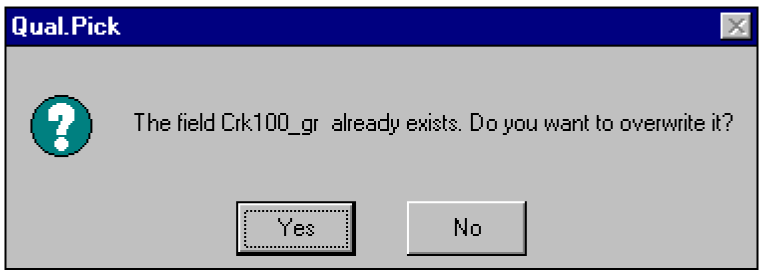


Figure 2.7: Overwrite an existing field (Qual.Pick)

Whether a point is in a creek can then be checked by looking at its value in the new field, which is either 1 or no data. In the second case, the location of the point must be modified so that it will eventually be located in the creek. The modification is done with the script Qual.Addpoint (Appendix C), customized with the tool . This tool is activated when the button is depressed. A new point is created by clicking on its desired location on the view. A message box gives the alternative between creating a new point coverage and modifying an existing one (Figure 2.8).

Figure 2.8: Create a new point coverage (Qual.Addpoint)

If the objective is to modify the stations' point coverage, the answer is “no”. Another message box prompts for the name of the coverage to modify (Figure 2.9).

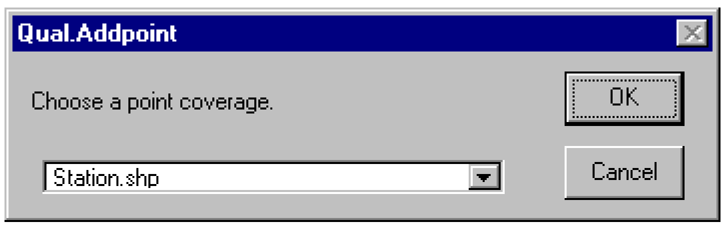


Figure 2.9: Choose the coverage to modify (Qual.Addpoint)

Once the point coverage has been chosen, the user is asked wether he wants to use the snap option. This option allows one to ascertain that a point is exactly located on a given line coverage representing the creeks, which is the vector representation of a considered grid stream network. This option is activated by answering “yes” to the message box shown in Figure 2.10.

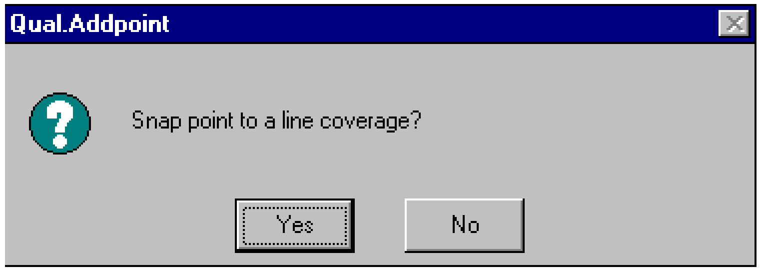


Figure 2.10: Snap option (Qual.Addpoint)

A message box prompts for the line coverage filename to use (Figure 2.11). The line coverage must correspond to the grid representation of the creeks. In this case, the coverage (*crk100_cv)* used is the creeks corresponding to a threshold, i.e. a drainage area, of 100 cells).

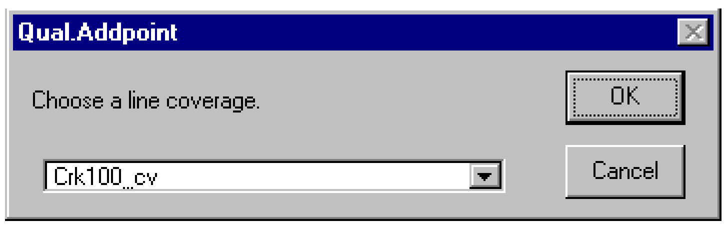


Figure 2.11: Choose a line coverage for the snap option (Qual.Addpoint)

The next message box tells the user wether the point has been successfully snapped to the chosen line coverage. If the procedure has been successful, the following message box appears (Figure 2.12).

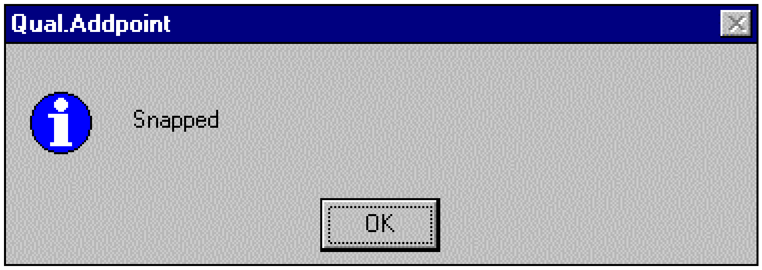


Figure 2.12: Successful snap (Qual.Addpoint)

However, if the distance from the point to the line is larger than the tolerance value, the point can not be snapped and the message “No line theme found” appears. The point remains at the location first chosen.

The user then has the option to edit the fields for the new point. For a new coverage, the only existing field is the field “id”. Other fields can be added by using the commands *Table/Start Editing* and *Edit/Add Field* in ArcView. For an existing coverage, a message box listing up to the first ten fields appears (Figure 2.13).

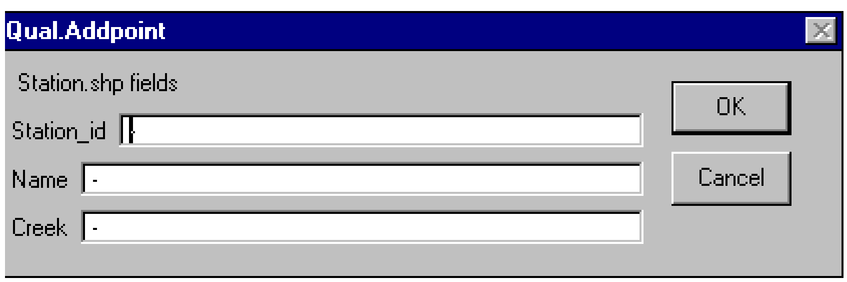


Figure 2.13: Edit the fields (Qual.Addpoint)

Figure 2.14 shows the location of a particular gage before and after the correction.

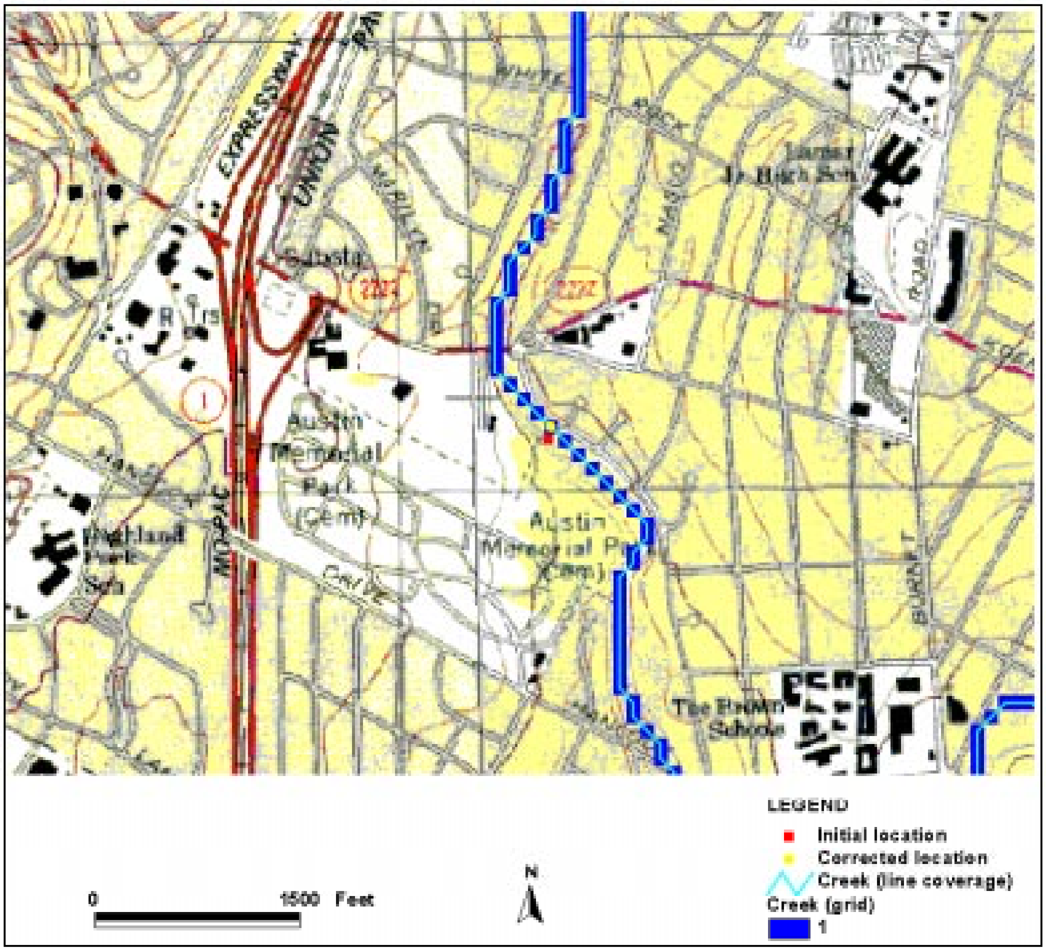


Figure 2.14: Gage location before and after correction

Once all the stations have been modified, the old points have become obsolete and must be deleted by highlighting their record in the attribute table and by using the commands *Table/Start Editing* and *Edit/Delete Records*.

The new point coverage must be checked as some points located on the line may also be at the junction of two cells, and hence have no data for grid value (Figure 2.15).

To avoid this problem, the grid can be used as a background theme when locating the points.

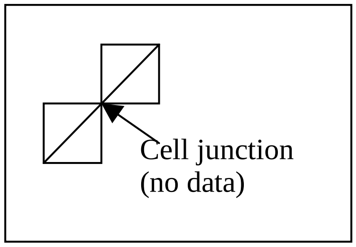


Figure 2.15: Cell junction

2.3 ENVIRONMENTAL INTEGRITY INDEX SITES

2.3.1 Locate the stations

Locating the Environmental Integrity Index stations required a different approach since their geographic coordinates were not available. At the beginning of the study, a Global Positioning System (GPS) was used to locate 45 Environmental Integrity Index sites. The same procedure used for the USGS stations was then applied to create the corresponding point coverage for those sites.

The City of Austin eventually provided a point coverage of new sites, which included the Environmental Integrity Index stations. However, there were additional stations to consider and two new sources of information became available, a scanned map and a detailed roads coverage (double line roads). This made it possible to locate the Environmental Integrity Index stations by direct comparison with a scanned map.

- **Scanned map**

Scanned map images of USGS topographic sheets for the Austin area were obtained from Horizons Technologies (http://www.horizons.com), Inc at a scale of

1:24,000. These map images were in geographic coordinates and had to be converted to State Plane coordinates. The projection file included a X and Y raster shift which was defined empirically by Francisco Olivera so that the streams defined from the 30m digital elevation model reasonably matched the blue line representation of the same streams on the USGS map image. The map was converted to a grid, which was projected according to the projection file shown in Table 2.7. It was finally reconverted back to a map.

Arc: **grid**
Grid: **imagegrid austin.tif aust_geo colors**
Grid: **project grid aust_geo aust_stp stamap_prj**
Grid: **gridimage aust_stp colors austinsp.tif TIFF**

Table 2.7: Scanned map projection file (stamap_prj)

Projection	STATEPLANE
Zone	5376
Datum	NAD27
Zunits	NO
Units	FEET
Spheroid	CLARKE1866
Xshift	112.0000000000
Yshift	-69.0000000000
Parameters	
End	

The result is a map (Figure 2.16) which can serve as a background to the other coverages or grids displayed.

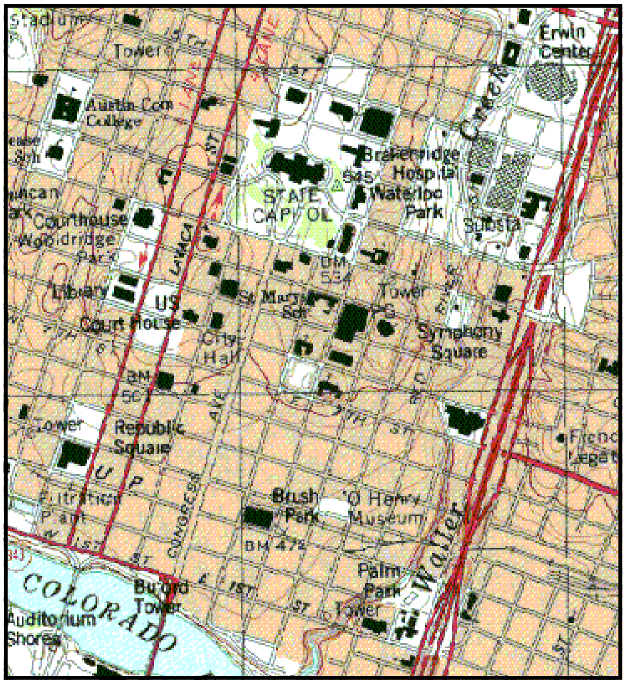


Figure 2.16: 1:24,000 USGS scanned map (detail)

The map can be used to directly locate the points representing the stations if their location is shown on the equivalent paper USGS topo sheet. It can also be used as a new source to confirm the location of the creeks.

- **Double line roads**

The double line road coverage also gives important information about the area (Figure 2.17). Combined with the scanned maps, this coverage enables one to locate the points more accurately, as the stations are often referenced using street name.

Figure 2.17: Double line roads coverage of Austin

2.3.2 Create a point coverage in ArcView

Since the locations of the Environmnetal Integrity Index stations were shown on a USGS map similar to the scanned map, it was possible to create the point coverage by clicking directly on the View window in ArcView at the desired location. The script used to create the point coverage is the same that was used to modify the USGS coverage previously (Qual.Addpoint, Appendix C). This script also guarantees that the points are located in the creeks.

The script is first used to create a new point coverage and then to add new points to it as was shown for the USGS stations. Hence the answer to the first message box, asking the user wether he, or she, wants to create a new point coverage is “yes”. The next message box prompts the user for the path and the name of the new point coverage (Figure 2.18).

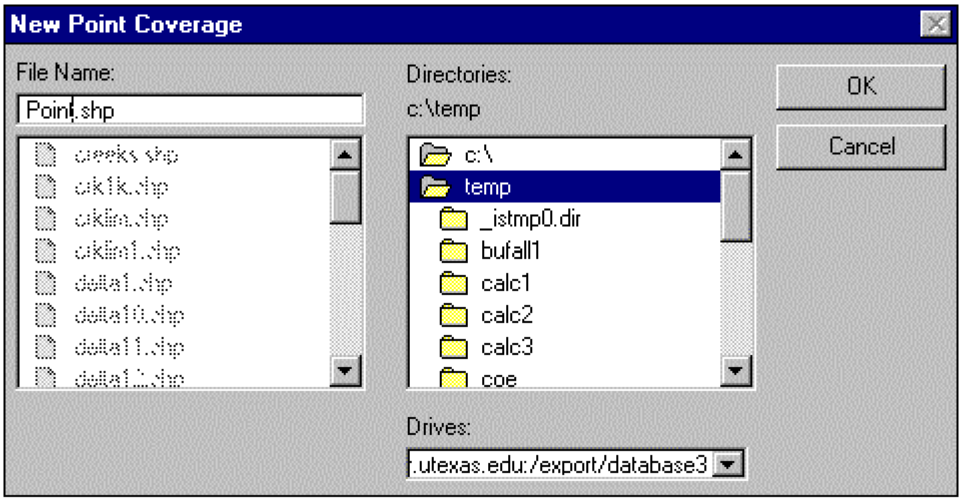


Figure 2.18: Name and path of the new coverage (Qual.Addpoint)

The field "id" is created automatically in the attribute table of the point coverage. Other fields can be added by editing the table by using the commands *Table/Start Editing* and *Edit/Addfield*. Other points are added by using the procedure described for the USGS stations.

2.4 RETRIEVE INFORMATION AT THE STATIONS

2.4.1 Retrieve the grid values

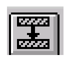
The goal of the project is to get the loads and watershed properties (impervious cover, discharge, percentage of land use...) at the Environmental Integrity Index sites. ArcView includes an information button  which enables the user to get the information from an attribute table for any location. This process is efficient when dealing with a limited number of points but it soon begins to be time-consuming for large numbers of points.

The script Qual.Pick presented in section 2.2.3 was written to allow all the information to be gathered in one step. This script allows the user to select a point coverage and one (or several) grid(s), and to retrieve the values of the grid(s) at the location of the points. The values are written in fields in the point coverage attribute table named by default after the corresponding grids. However, if a field already exists, a message box gives the user a choice between renaming the field and overwriting the already existing one.

2.4.2 Characterize the stations

Information concerning the stations can be gathered and written in Excel files. To be used in Arc/Info or ArcView, the data must be converted either to a .dbf or a .txt format, by using the option *File/Save as* in Excel.

In Arc/Info Tables, some data related to a coverage can be permanently added to the attributes table of the coverage. In ArcView however, the *join* function enables one to edit the attribute table with the new data only in that project. The original attribute table is not modified. Since it is more convenient to work with tabular data in ArcView, a script was written to allow two tables to be permanently joined in ArcView.

Without a script, the more convenient way to add data contained in one table to a corresponding attribute table in ArcView is to join the two tables, then to add new fields corresponding to those which have just been joined and set the new field values equal to the joined field values. The last step is to remove the join, which yields an attribute table with only the manually added fields. The same approach was used in the script Qual.Join, which just automates this process (Appendix C). The script can be run by clicking on the button  which appears when a table is active. Four message boxes prompt the user respectively for:

- a destination table (Figure 2.19)
- a common field for the destination table
- a source table
- a common field for the source table

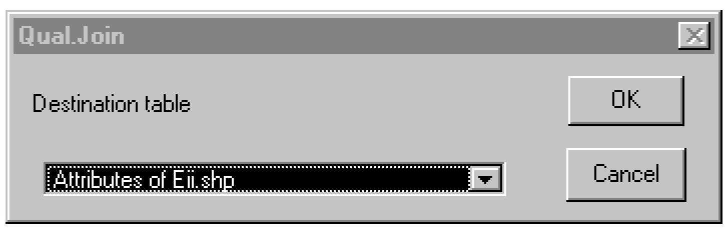


Figure 2.19: Choose a destination table (Qual.Join)

The destination table is the table the fields are joined to. The common field is the field used to relate the records of the destination and source tables.

2.4.3 Delete the fields

Data retrieval and table joinS may create numerous fields in a table. When the field are no longer used, it is wise to delete them. The delete command in ArcView (*Table/Start Editing*, then *Edit/Delete Field*) enables one to delete one field at a time. Given the large number of fields which may have to be deleted, the script Qual.Delete

(Appendix C) was written to allow several fields to be deleted in one step. This script is activated with the button , which is visible when a table is active. A message box appears by clicking on the button, prompting the user for the table to edit (Figure 2.20).

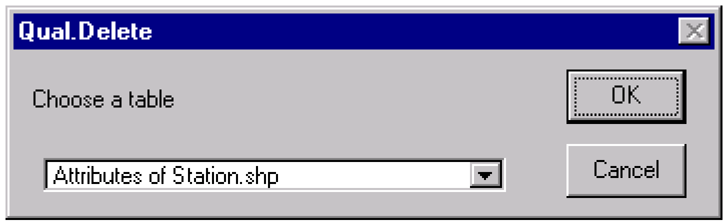


Figure 2.20: Choose a table to edit (Qual.Delete)

A second message box prompts for the list of fields to delete (Figure 2.21). Note that some of the fields are read-only fields (e.g. shape) and cannot be deleted.

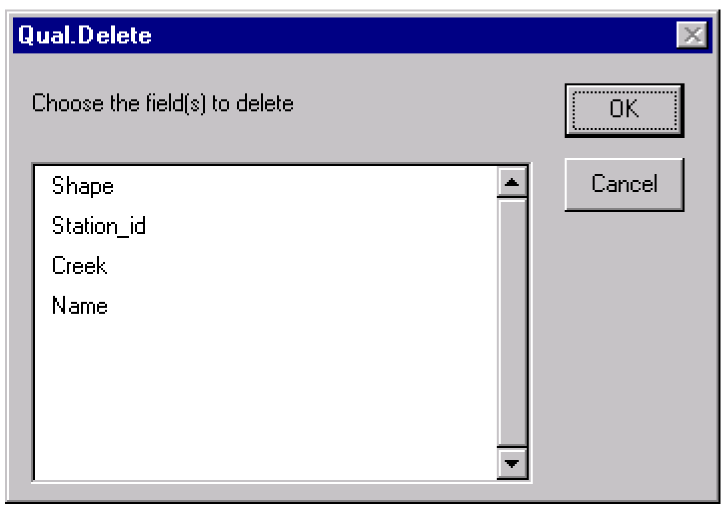


Figure 2.21: Choose the fields to delete (Qual.Delete)

After confirmation that the fields chosen are really the ones which must be deleted, the deleting operation takes place.

Chapter 3: GIS topography characterization

3.1 DIGITAL ELEVATION MODEL (DEM)

3.1.1 30m DEM

The study area includes Austin and its immediate surroundings, and can be represented by twenty-seven 7.5’ USGS quadrants, each of them corresponding to an elevation grid called Digital Elevation Model (DEM) (Figure 3.1). Each cell of the grid is assigned the average elevation of the area represented by the cell. Each quadrant is composed of 466 rows and 406 columns: the total area in Figure 3.1 is represented by about 5 million 30-meter cells and covers about 4,600 km^2.

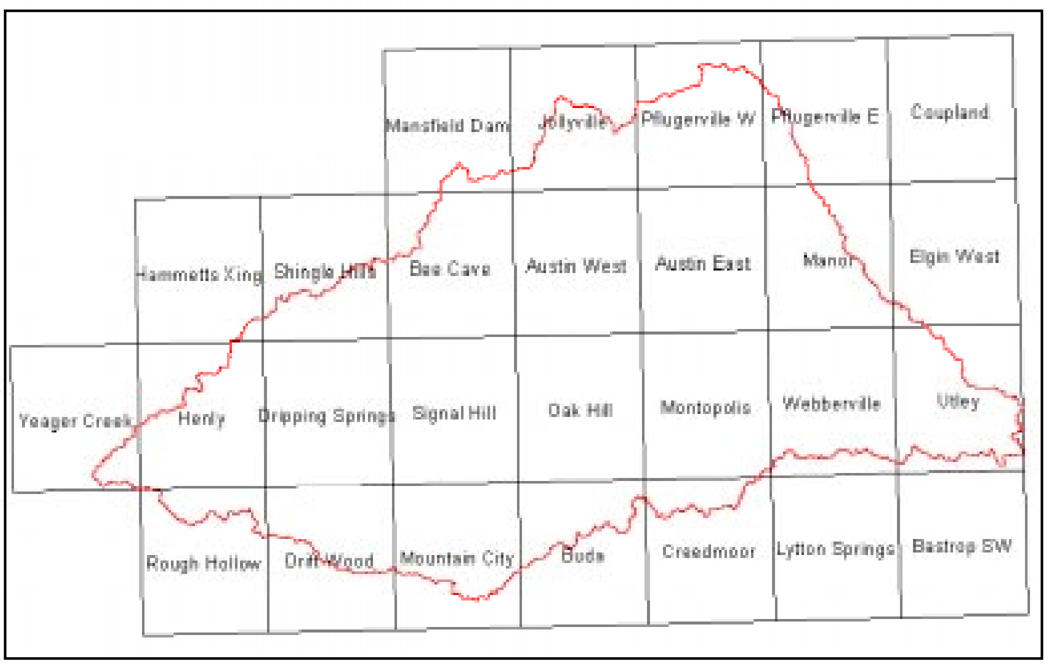


Figure 3.1: USGS 1:24,000 quadrant sheets

The study area (outline in red in Figure 3.1) covers an area of 2,300 km^2.

Digital Elevation Models consist of a sampled array of elevations for ground positions at regularly spaced intervals. They are used to define the direction of the flow according to the topography and the drainage areas for any cell. Francisco Olivera did a study of Austin based on 90m DEMs (Olivera, Maidment and Charbeneau, 1996). When this study was done, 90m was the finest cell size available for the whole area. The results show that the delineation is not good at all for some of the drainage areas. For example, there is a 44% relative error between the areas observed on the field (about 5.7 mi^2) and delineated (8.2 mi^2) for Waller Creek, which is a small urban watershed (Figure 3.2). Moreover, the differences are more important at the gages located in the watershed (Table 3.1).

The use of 30m DEMs, which were available at the time for only part of the area, greatly improves the delineation. Table 3.1 shows the different values obtained (from the USGS and by delineation) for the drainage areas of the USGS stations located within the Waller Creek watershed. The maximum relative error is less than 5%. Figure 3.2 shows the differences between the digitized watersheds and those based on 30m or 90m DEMs for Waller Creek.

Table 3.1: Watersheds areas comparison for Waller Creek

Stations (Waller Creek)	38^{th}	23^{rd}
USGS	2.31 mi^2	4.13 mi^2
30m DEM	2.39 mi^2	4.33 mi^2
Relative error (30m)	3.5%	4.8%
90m DEM	5.18 mi^2	6.65 mi^2
Relative error (90m)	124%	61%

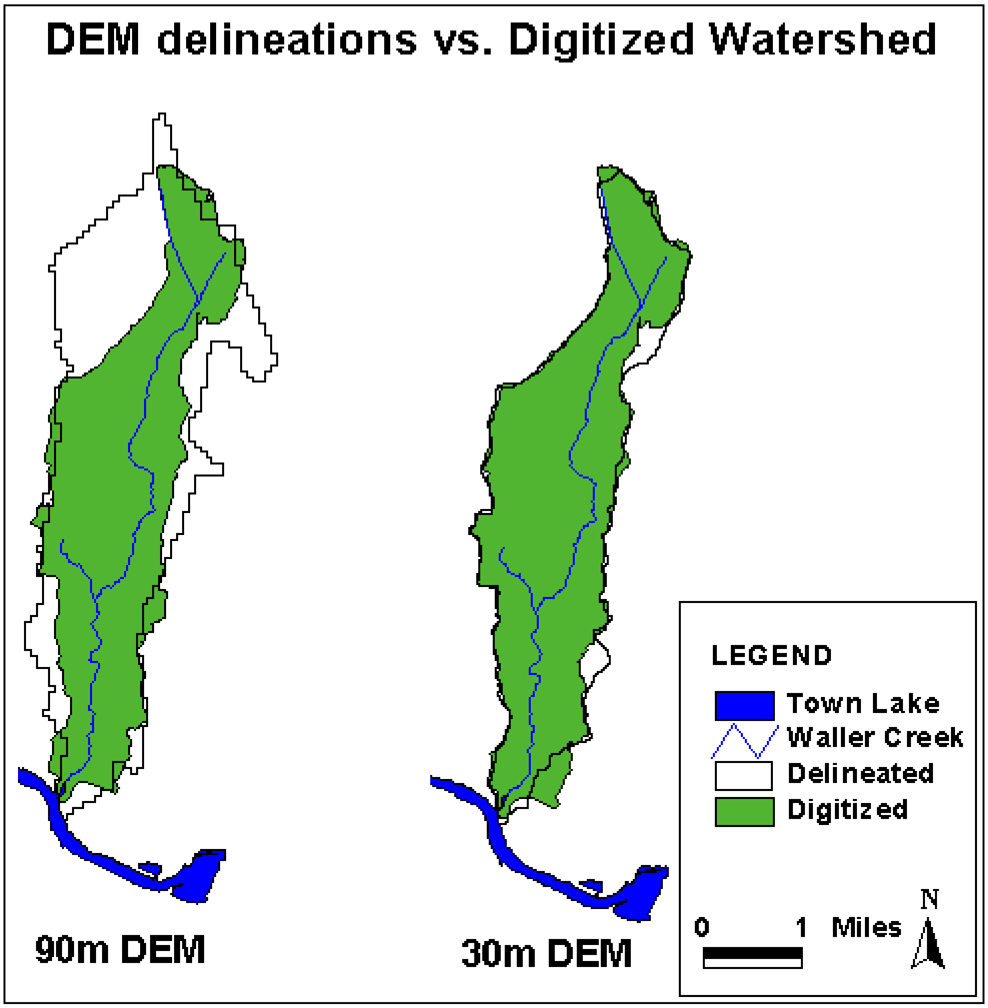


Figure 3.2: Delineated Watershed vs. Digitized Watershed (Waller Creek)

The comparison between the areas defined by the 30m and 90m DEMs underlines the fact that the difference of cells size (90m to 30m) enables one to greatly improve the delineation.

It was very important to define as accurately as possible the drainage areas corresponding to the USGS gauged stations as their data are used to build the model.

There was a need to improve the delineation based on the 90m DEMs. In parallel with this project, the City of Austin worked with a contractor to provide 30m DEMs for the whole drainage area. The new availability for the whole Austin area of 30m DEMs which enables to improve ten times the accuracy (Figure 3.3) allows one to really consider the use of GIS for city-scale modeling. Each DEM being composed of about 190,000 cells, the total area is represented by more than 5 million cells.

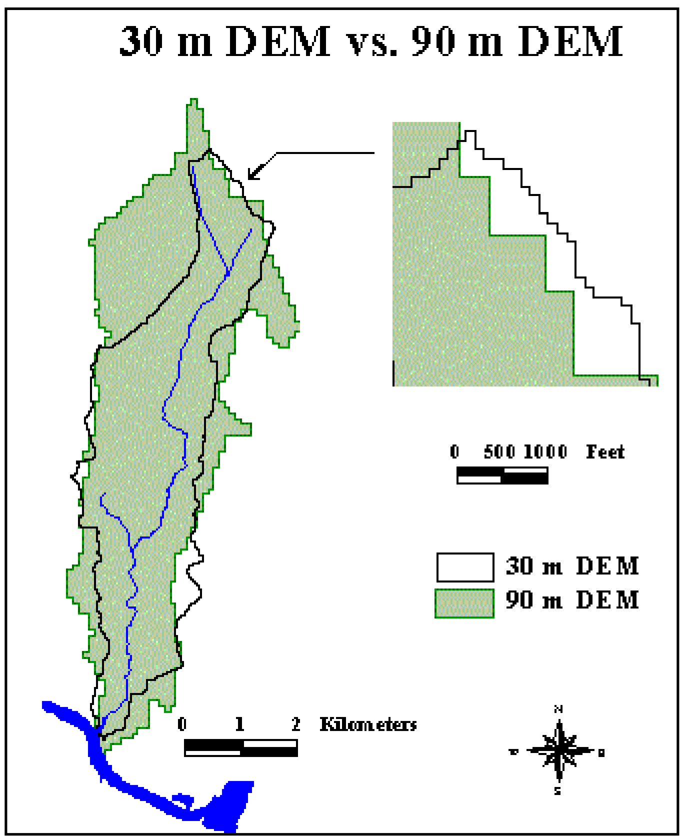


Figure 3.3: Comparison between 30m and 90m DEMs

While the comparison between the delineation obtained with the two types of DEMs shows the greater accuracy of the 30m DEMs because of the smaller cell size, there are however two main drawbacks to their use:

- they are more disk storage consuming.
- the computations take longer: for 5 million cells, the execution time for a flowdirection or a flowaccumulation function is about one hour.

3.1.2 Use of a DEM

A DEM is a representation of the topography of a given area. It is used to create two further grids (Figure 3.4):

- a Flowdirection grid, which indicates the path of the water through the landscape. The cell value corresponds to one of the eight geographic directions (north, northwest...) indicating the direction of steepest slope.
- a Flowaccumulation grid, which defines the size of the drainage area. Each cell as its value the number of cells located upstream, whose drainage passes through the given cell.

A variation to the Flowaccumulation is to associate a value or “weight” (e.g. runoff produced in a cell) to each cell and to sum this value instead of summing the number of cells. This concept is the backbone of the model: once the cells have been characterized, it is straightforward to obtain the sum of the characteristics values of the upstream cells.

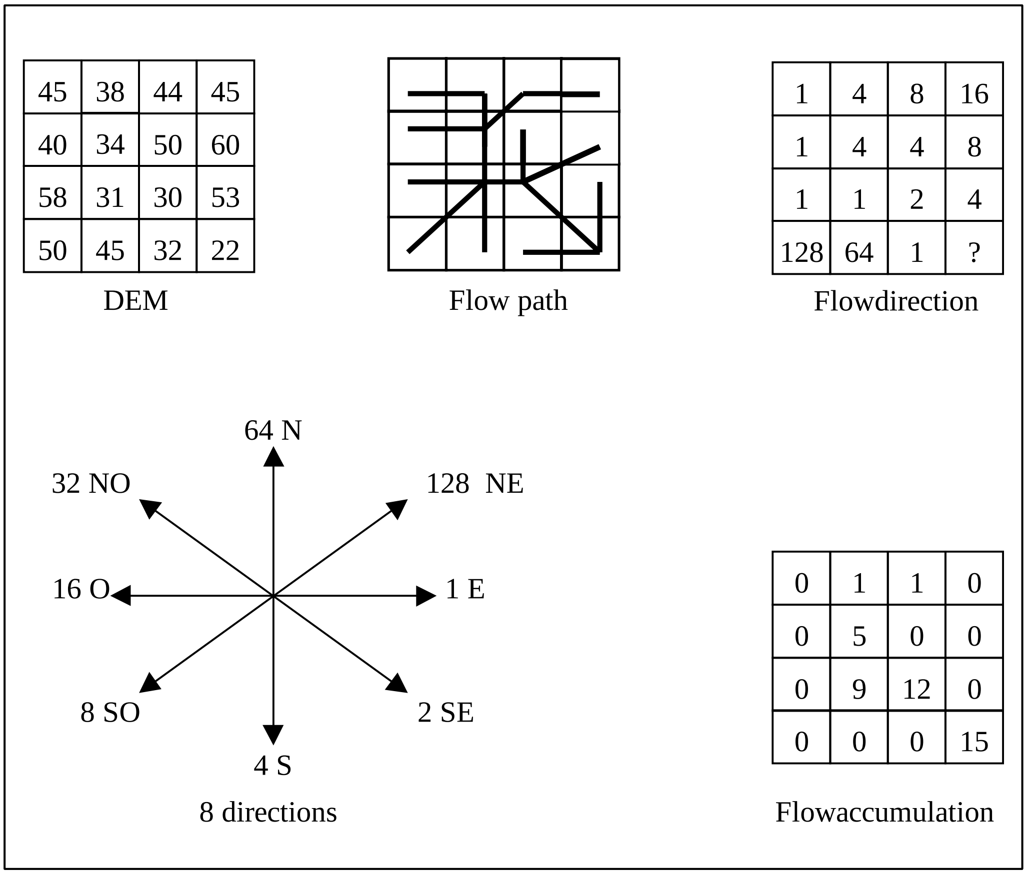


Figure 3.4: Flowdirection and Flowaccumulation grids

3.2 DEM PROCESSING

3.2.1 Pre-processing

Twenty-seven USGS 7.5'quadrants are needed to cover the total area, each of them being represented by a different DEM. In their initial format, the DEMs (source: Vernon F. Meyer and Associates, Inc) cannot be used in GIS. They have to go through two pre-processing steps, to be reblocked and to be converted from USGS format to Arc/Info lattice or grid format.

%**dd if=input_file of=output_file cbs=1024 conv=unblock**
%**arc**
Arc: **demlattice output_file output_dem**
(Convert a DEM in USGS or TAME format to a lattice)

The resulting files can now be displayed either in Arc/Info or ArcView.

3.2.2 Merging/Projection

The following two steps consist of merging all the quadrants together and projecting the resulting DEM into the state plane coordinate system.

- **Merge the grids**

In Arc/Info, each command must be typed within a few lines. Given the important number of grids to merge, it was necessary to proceed in two steps.

Grid: **alldem = merge (austn_e , austn_w , bastrpsw , beecave, buda , coupland , creedmor , drift , dripsprg, elgin_w , hamcross , henly , jollyvil , lyttsprg , manfiel , manor , montop , mountcty , oakhill , pflug_e , pflug_w , roughhol)**

Grid: **alldem1 = merge (alldem , shinghhi , signalhi , utley , webber , yeager)**

- **Project to state plane**

The resulting DEM is in the UTM (Universal Transverse Mercator) projection system. It needs to be projected into the state plane projection system. The datum NAD27 used in the project is based on the foot as the length measure. The 30m cell size is hence converted to a 100ft cell size and all the elevations values are resampled onto the new grid during the map projection process.

Grid: **dem_st = project (alldem1 , # , # , 100)**
Project: **output**
Project: **projection state**
Project: **zone 5376**
Project: **units feet**
Project: **parameters**
Project: **end**

3.2.3 Eliminate the no data cells

As the borders of the grids are not perfectly adjusted, some no data cells are created at the quadrant borders during the merging process. These cells can be viewed by displaying the projected DEM in ArcView. It is very important to eliminate these no data cells since otherwise they would act as sinks for the water. These cells must be assigned a value, the average value of the surrounding cells, by editing the DEM grid dem_st using Arctools. The procedure to check that all no data cells have been eliminated consists of focusing on different zones of interest within the area where there should not be any no data cells beside the ones due to the merging process (quadrants borders). For each zone studied, the no data cells are assigned the value zero, as there is no zero in the original grid. Then a grid can be created, with value one at the no data cells and no data elsewhere. The number of remaining no data cells is obtained by describing the grid in Arc/Info.

Grid: **xxx = con (isnull (dem_st) , 0 , dem_st)**
Grid: **yyy = con (xxx == 0 , 1)**
Grid: **list yyy.vat**

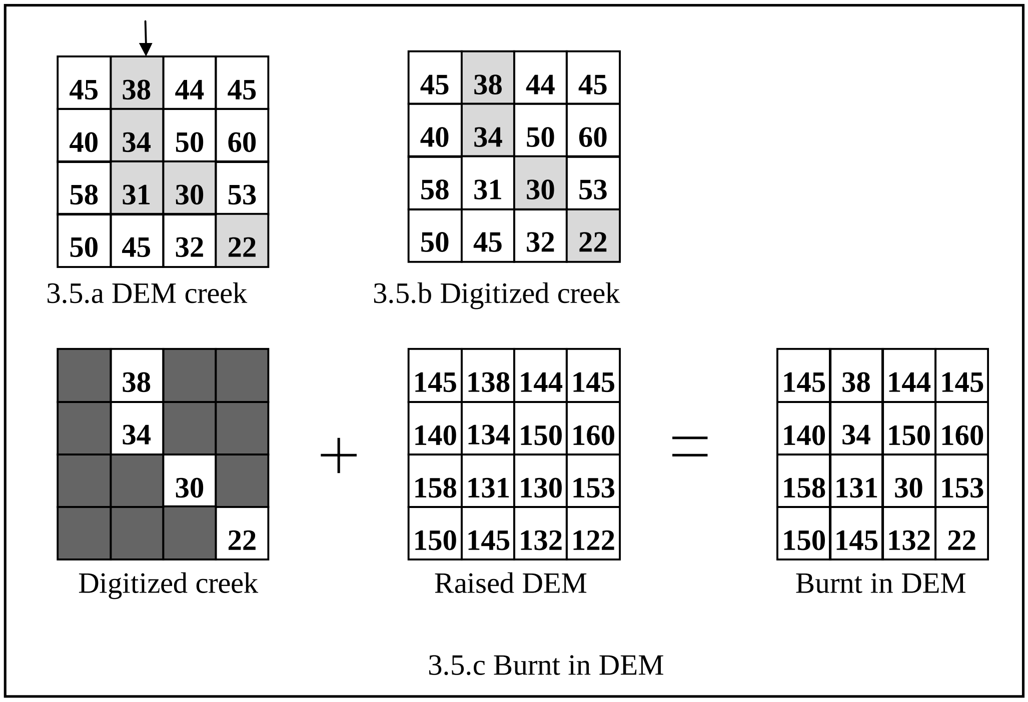
Record	VALUE	COUNT
1	1	5

In this case, there are 5 no data cells left. The editing process is carried on until the average value of the surrounding cells has been assigned to each no data cell.

3.2.4 Burn in process

The goal of the study is to determine the non-point source pollutant loads in Austin creeks. So it is important that the delineated creeks match the observed creeks. To guarantee that, the creeks are imbedded in the grid through a "burn in" process. A large elevation value is added to the elevation grid (10,000 ft in this case), so that any cell of the new grid has an elevation higher than the maximum elevation in the original grid. A grid containing only the creeks at their initial elevation value is then merged with the raised DEM so that the resulting DEM has a thin, deep trench at the stream locations and a normal landscape elsewhere.

The burn in process is important because the real creeks do not always correspond to the creeks defined by the DEM. For example, assuming that a creek is coming into the cell with elevation 38 in Figure 3.5.a, according to the DEM it is then going to the cell located in the direction of steepest descent. The corresponding flow path is shown by the shaded cells in Figure 3.5.a. In reality, the true creek trajectory may be different (Figure 3.5.b). The "burn in" procedure is applied to guarantee that the "correct" trajectory is obtained (Figure 3.5.c).



3.5.a DEM creek

3.5.b Digitized creek

Digitized creek

Raised DEM

Burnt in DEM

3.5.c Burnt in DEM

Figure 3.5: Burn in process

This procedure can be done either in Arc/Info grid or in ArcView. The Arc/Info method is presented here. The method in ArcView is available on the internet (Olivera, 1996).

- **Complete the stream coverage**

As the digitized creek coverage provided by the City of Austin does not cover the whole study region, it must be augmented with streams from another source, which is the EPA Reach File 3 (RF3). For Austin, those files are 12090205 (*205_st*) and 12090301 (*301_st*).

There are two solutions to augment the streams:

- convert the coverages to grids, use the merge command to combine the two creek coverages, and then make the appropriate corrections in Arctools.
- proceed directly with the coverages, make the corrections in Arcedit and use a script written by Zichuan Ye (Qual.Mergethm, Appendix C) in ArcView to merge them as there is no command in Arc/Info to merge arc coverages.

As the conversions to grid and then back to a coverage modify slightly the coverages, to obtain a creek coverage as close as possible to the original, the second method was implemented. The part of the RF3 to be used is selected in ArcView, converted first to a shapefile and then to an arc coverage called *rf3_cplt*. Since the coverages *creeks* and *rf3_cplt* do not match at their junction, *rf3_cplt* is corrected in Arc/Info Arcedit.

Arc: **arcedit**

The first steps consist in defining the edit coverage, the edit feature (arcs in this case), the back coverage and the back environment as well as the drawing environment.

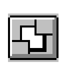
Arcedit: **ec rf3_cplt**
Arcedit: **ef arcs**
Arcedit: **backenvironment arc**
Arcedit: **backcoverage creeks 2**
Arcedit: **drawenvironment arc**
Arcedit: **mape rf3_cplt**
Arcedit: **&station 9999**
Arcedit: **draw**

Since arc is the edit feature, only arcs can be deleted. Hence if the line to be deleted is just a part of an arc, it must be first defined as an arc by using the command *split*, and then deleted.

Arcedit: **select ***
Arcedit: **split**
Arcedit: **select ***
Arcedit: **delete**
Arcedit: **save**

As the coverages come from two different sources (City of Austin and EPA), they do not match exactly. Some links must be established between them.

Arcedit: **ec rf3_cplt**
Arcedit: **ef link**
Arcedit: **backcoverage creeks 2**
Arcedit: **backenvironment arc node**
Arcedit: **drawenvironment arc node link**
Arcedit: **mape rf3_cplt**
Arcedit: **&station 9999**
Arcedit: **draw**
Arcedit: **snapcoverage creeks**
Arcedit: **linkfeatures node node**
Arcedit: **snapping closest ***
Arcedit: **limitautolink box**
Arcedit: **autolink**
Arcedit: **adjust**
Arcedit: **save**

The coverages *rf3_cplt* and *creeks* must then be joined in ArcView by using the script *Qual.Mergetheme* ( or *QualTools/Merge Themes*) to create *burn_crk*, the stream network used in the burn in process. In fact a grid form of the water bodies, which takes into account both lakes and creeks, is used in the process.

- **Burn in**

The first step is to convert the digitized network of lakes and creeks into two grids in State Plane coordinates, and merge them together to create a grid of value one.

Grid: **lakes_gr = polygrid (lakes)**
Grid: **creek_gr = polygrid (burn_crk)**
Grid: **str_gr = merge (lakes_gr , creek_gr)**
Grid: **water1_gr = con (str_gr , 1)**
Grid: **water_gr = water1_gr * dem_st**

Gaps in the stream network are corrected in Arctools, because one of the following procedures consists in filling the sinks. A gap in a creek prevents the water from flowing and this creek will be considered as a sink and filled. Finally, the water grid (water_gr) is merged with the DEM to which 10,000 ft has been added.

Grid: **dem_pls = dem_st + 10000**
Grid: **burn_dem = merge (water_gr , dem_pls)**

The order of the terms in the merge command is important. The values of the second grid will be taken only for the cells in which the first grid has no data. The result of the command *merge(dem_pls , water_gr)* is then dem_pls.

3.2.5 Burning walls

In some areas, the DEM delineated watersheds may not match the “real” watersheds, because the zone is too flat or because of storm sewers which divert the runoff. The City of Austin was concerned with these differences, especially for Blunn and West Bouldin watersheds. A procedure was created so that the watershed boundaries can be forced to be where they are supposed to be (same concept as the burn in process).

Instead of digging trenches, walls are built. The coverage *wsheds* being the polygon coverage of the "real" watersheds, the procedure is:

Arc: **build wsheds line**
Arc: **grid**
Grid: **wshed_gr = linegrid (wsheds)**
Grid: **wall1 = con (wshed_gr , 1)**
Grid**: wall2 = wall1 * dem_st**
Grid: **wall20k = wall2 + 20000**
Grid: **burnw_dem = merge (water_gr , wall20k , dem_pls)**

This procedure affects the original topography: the "forced boundaries" must be at the highest elevations in the watersheds. Since the water must flow from the boundaries toward the creek, the elevations of the cells located between the original and the "forced" boundaries must also be modified through a filling process (section 3.2.6). The information given by the DEM is not used anymore. As the study is based on the topography, this procedure was eventually not used.

3.2.6 Flowdirection/Flowaccumulation computation

The DEM on which the creeks have been burnt in must undergo one last procedure before being used to create the flowdirection and flowaccumulation grids. If a cell is surrounded by higher elevation cells, the water is trapped in that cell and can not flow. It can really happen if, for example, the water infiltrates through the bottom of a lake. In the case studied here however, these cells must be corrected by using the function fill, which modifies the elevation values to eliminate these problems. As this function is based on the direction of the flow, it enables one to create also the flowdirection grid, *burn_fdr*.

Grid: **fill burn_dem burn_fil # # burn_fdr**

The flowaccumulation grid is computed from the flowdirection grid.

Grid**: burn_fac = flowaccumulation (burn_fdr)**

3.3 VALIDATION OF THE METHOD

It is necessary to check the validity of the chosen approach for the entire area. The model is correct only if the DEM gives an adequate representation of the topography.

3.3.1 Quantitative assessment of the topographic representation

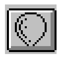
With the flowaccumulation grid, the drainage basin of any cell of the grid can be determined. The areas of the basins corresponding to the USGS stations can be obtained from the USGS web site (http://txwww.cr.usgs.gov/cgi-bin/txnwis). By delineating the basins, a comparison between the areas can be done (Figure 3.6). The watersheds can be delineated either in Arc/Info or ArcView.

In Arc/Info, the delineation is based on a grid representation of the USGS stations point coverage.

Grid: **setcell 100**
Grid: **usgs_gr = pointgrid (usgs_cv**)

The USGS watersheds are then delineated using the watershed function and converted from a grid to a coverage:

Grid: **usgswshd_gr = watershed (burn_fdr , usgs_gr)**
Grid: **usgswshd_cv = gridpoly (usgswshd_gr)**

The delineation in ArcView uses the script *Qual.Watershed*, customized with the button  and the menu *QualTools/Watershed* (Appendix C). A series of message boxes prompt the user for the names of the output watershed grid and coverage, as well as for the input point coverage and flow direction grid. The last message box prompts the user for a ranking field. This field is used to identify the different points and their corresponding watershed. The values in that field must be integer numbers different for each record. They also must be within a range of 1,000,000 to be displayed in ArcView (integer lookup table range).

The drainage areas match quite well except in one case (station 8159000, Onion at US 183) where it appears that there is a problem of definition regarding what should be considered as the watershed. From a quantitative point of view the match of areas is good.

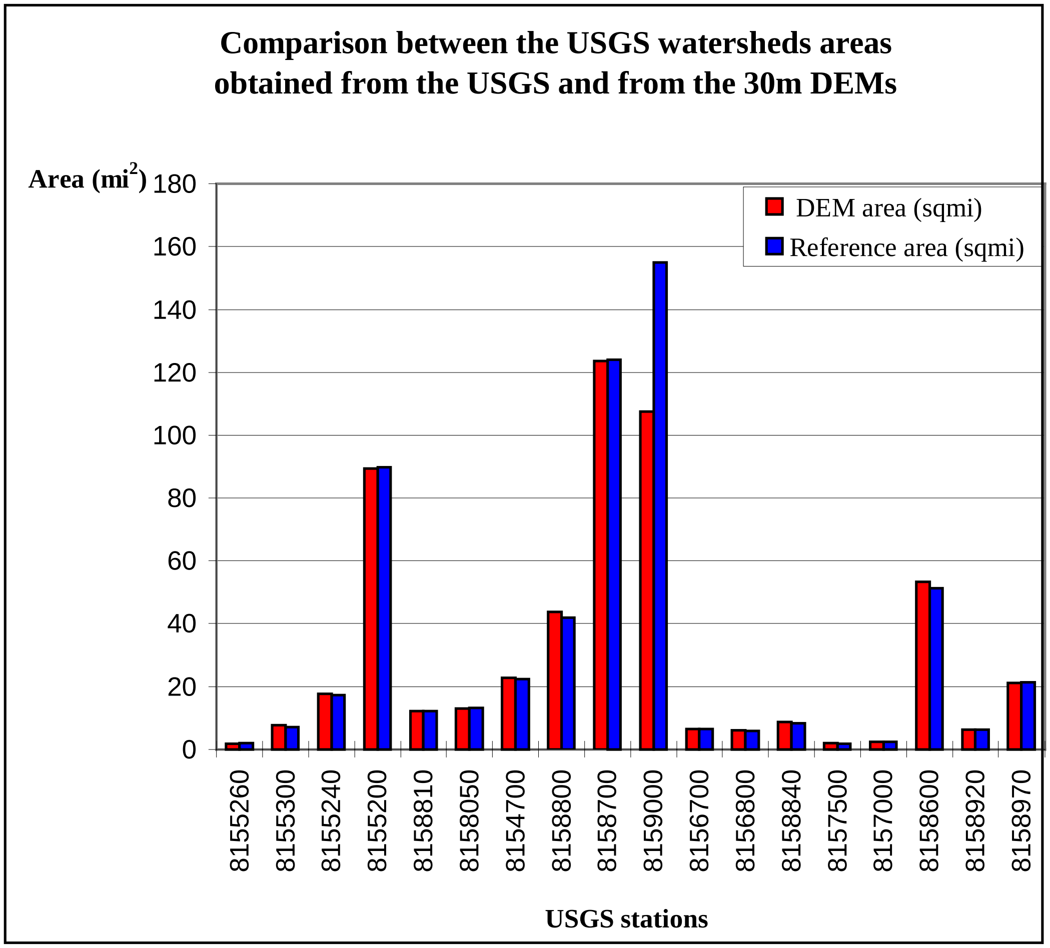


Figure 3.6: Comparison between the USGS stations drainage areas obtained from the USGS and delineated with the DEMs

3.3.2 Qualitative assessment of the topographic representation

It is not enough that the area values match. The delineated basin boundaries must equally correspond to the real ones. Digitized watersheds, based on field observations, are available for the Austin region. A comparison with DEM delineated watersheds shows that the model gives good results, except for some flat areas (Figure 3.7).

However, the results are good enough for the watersheds under study to carry on this approach.

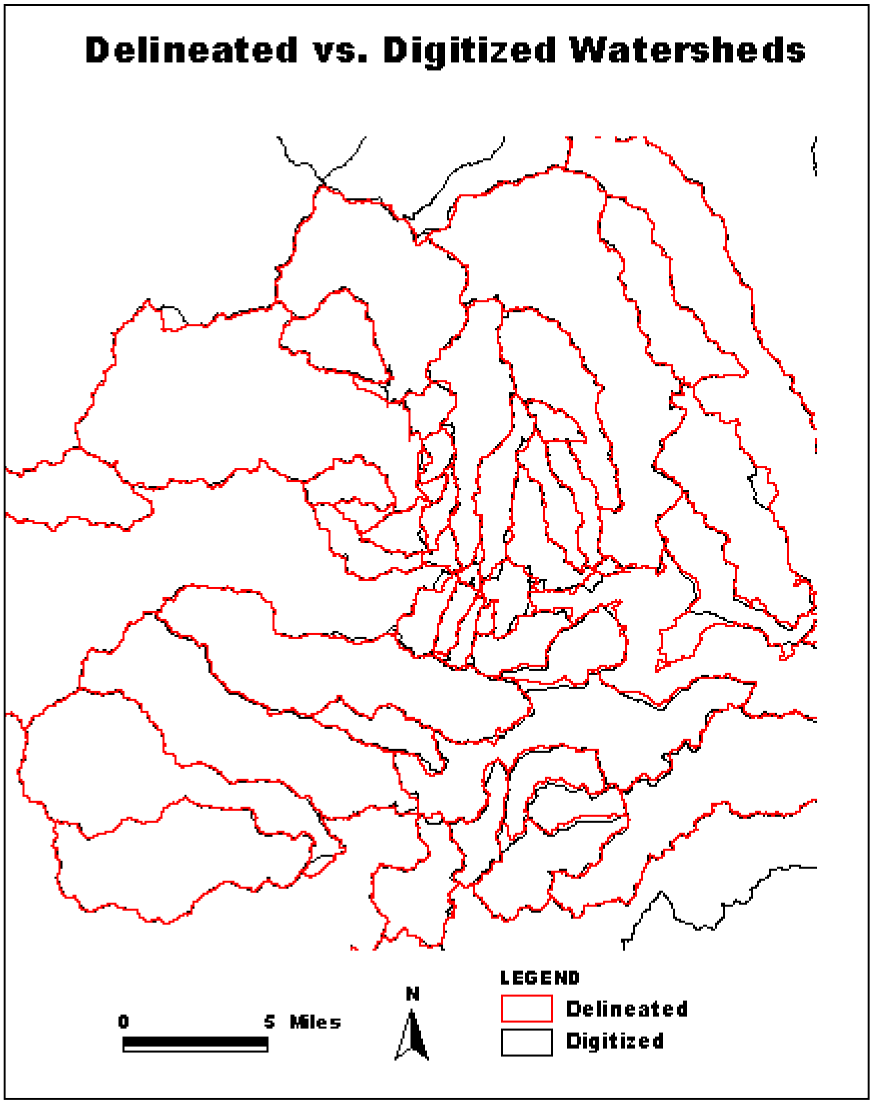


Figure 3.7: Comparison between digitized and delineated watersheds

3.4 CREEK AND WATERSHED DELINEATION

3.4.1 Method

A creek can be defined according to a flow accumulation value: the cells whose drainage area is bigger than a given threshold are considered to be in the creek.

Grid: **crk100_gr = con (burn_fac > 100 , 1)**
Grid: **crk100_cv = streamline (crk100_gr , burn_fdr)**
Grid: **lk100 = streamlink (crk100_gr , burn_fdr)**
Grid: **wshd100_gr = watershed (burn_fdr , lnk100)**
Grid: **wshd100_cv = gridpoly (wshd100_gr)**

The conversion from grid to arc coverage uses the function streamline instead of the function gridline. Gridline does not allow parallel lines if they are too close and creates instead a node (Figure 3.8).

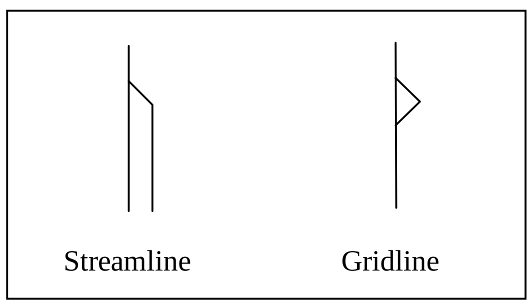


Figure 3.8: Difference between the commands streamline and gridline

The creeks can also be delineated in ArcView by using the script *Qual.Creek* (Appendix C). This script can be run by clicking on the button  or by using the command *QualTools/Creek*. The user is prompted for a flowdirection grid and a flowaccumulation grid, and for a creek threshold (number of 100ft cells). Two files are created: a creek grid and the associated line coverage.

Several creeks and watersheds networks can be used corresponding to different degrees of accuracy. Figure 3.9 represents the creeks delineated threshold values ranging from 100 to 10,000 cells.

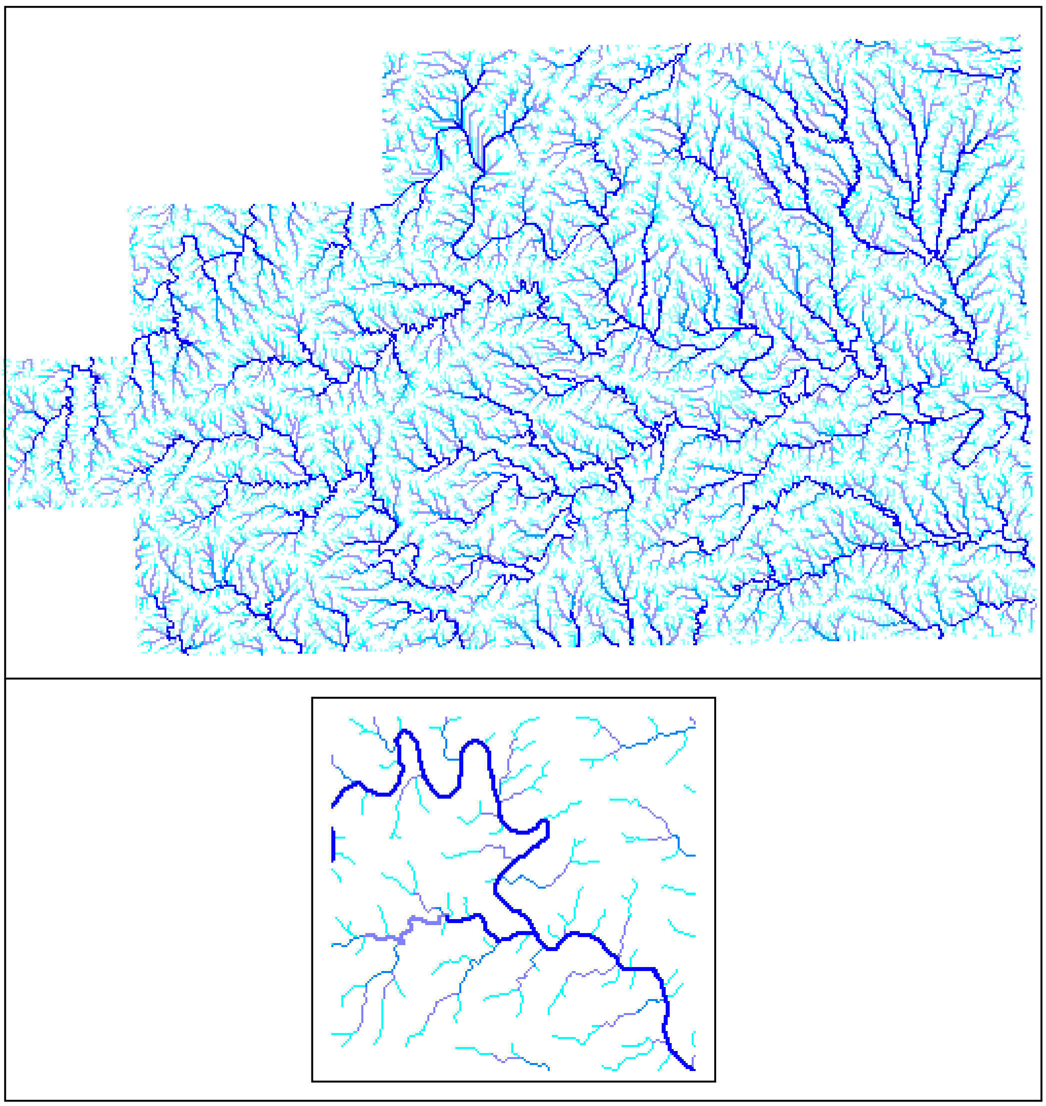


Figure 3.9: Stream network for different thresholds (100 to 10,000 cells)

It is possible to compare the location of the delineated creeks with the stream network of the scanned 1:24,000 USGS maps. The results are correct, except in a few cases for which the digitized creeks do not correspond to the streams on the map.

3.4.2 Define the upstream limit of a creek

A creek can be defined by a threshold value. All cells whose drainage area is bigger than a given value are considered to be in a creek. The upstream limit of the stream network can hence be found by looking for the first cells in the different creeks where the drainage area is greater than that threshold value. Each cell is 100ft*100ft = 10,000 ft^2 = 0.2296 acres in area.

For example, for a threshold of 64 acres (279 cells), the procedure is:

Grid: **crk279 = con (burn_fac > 279 , 1)**
Grid: **fac279 = flowaccumulation (burn_fdr , crk279)**
Grid: **pt279 = con (crk279 == 1 and fac279 == 0 , burn_fac)**
Grid: **pt279_cv1 = gridpoint (pt279)**
Grid: **project cover pt279_cv1 pt279_geo**
Grid: **addxy pt279_geo**
Grid: **project cover pt279_geo pt279_cv**
Grid: **crk279_cv = streamline (crk279 , burn_fdr)**

The result is a point coverage (Figure 3.10) representing the upstream limit of the streams defined as the point with a drainage zone of at least 64 acres (279 cells).

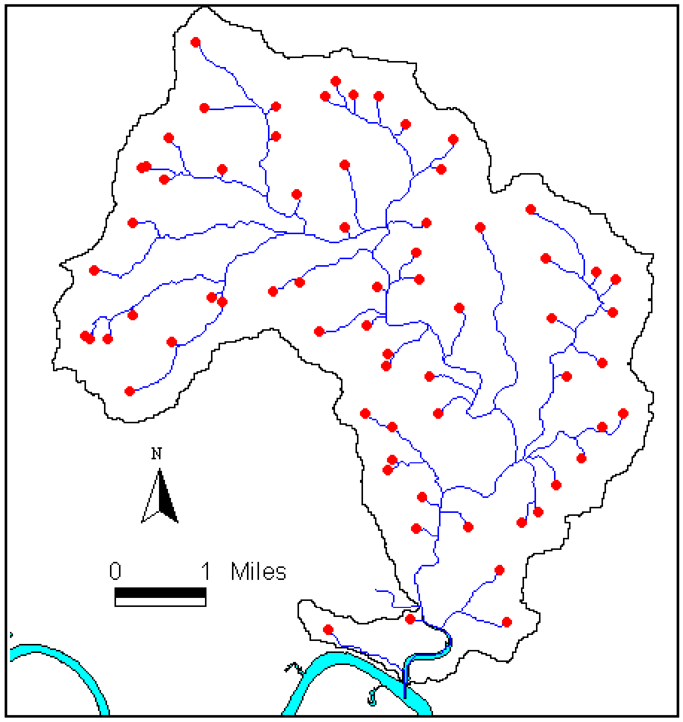
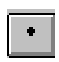
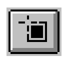


Figure 3.10: Creeks draining at least 64 acres in Bull Creek Watershed

An ArcView script (*Qual.Creeklimit*, Appendix C) allows the user to build a grid containing the points representing the limit of the creeks for a chosen threshold. This script can be run by clicking on the button  or by using the command *QualTools/Creek limit* when a View is active. The user is prompted for a flowaccumulation grid and a flowdirection grid, and for a creek threshold (number of 100ft cells).

3.4.3 Getting rid of the dangling polygons

A comparison between the watershed grid and the polygon coverage produced by the *Gridpoly* function shows that some "extra" watersheds (dangling polygons) appear during the conversion to the vector form. They are small polygons (a few cells) which have the same grid-code as a larger polygon with which they share a corner, but not a side. They are listed as separate records in the polygon attribute table.

A script to eliminate the dangling polygons was written by Zichuan Ye as part of an ArcView based watershed delineation application developed by Environmental Systems Research Institute (ESRI, Inc.) for the Texas Natural Resource Conservation Commission (TNRCC). This script (Qual.HydroZdslv) allows one to change the grid-code in dangling polygons to that of an adjacent polygon that does share a side. The script is customized with the button  and the command *QualTools/Dangling*. The user is prompted to enter the name of a polygon coverage (e.g. *wshd*) and of the associated polyline coverage. Both may have the same name. After running the script, the "dissolve" function in Arc/Info allows the user to merge the dangling polygons with the polygons with the same grid-code.

Arc: **dissolve wshd wshd_diss grid-code**

Some dangling polygons may remain in two cases:

- a no data cell is inside a watershed: the solution is to set its grid-code to the watershed grid-code.
- a cell with a grid-code among no data cells: the solution is to set its grid-code to a specific value corresponding to no data (i.e. –9999).

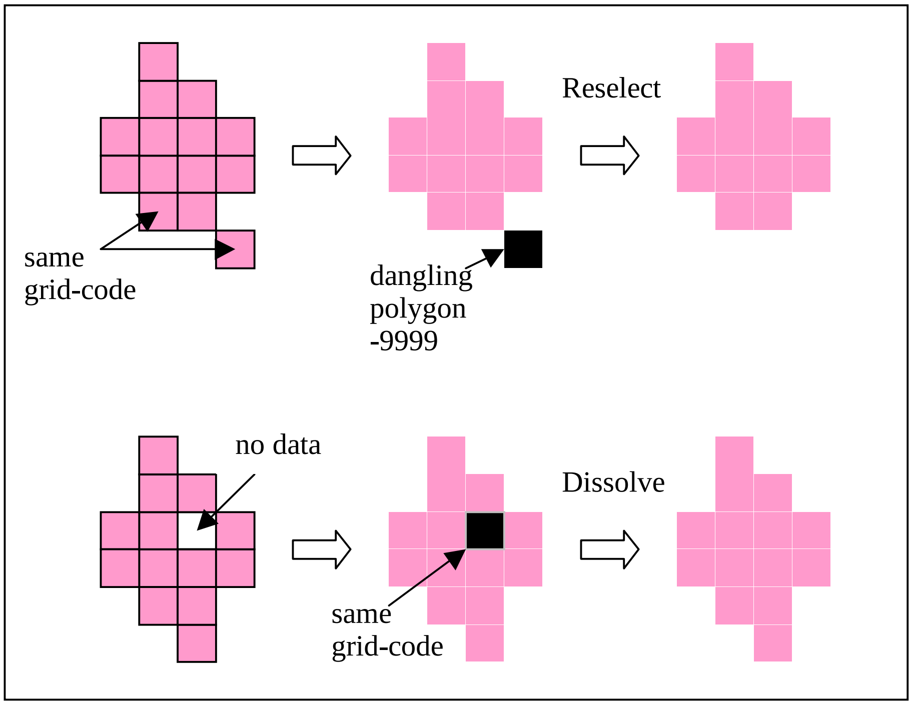


Figure 3.11: Dangling polygons

The modifications are made manually by editing the polygon attribute table in ArcView. For example, if *wshd_diss* is the watershed coverage obtained after running the script, its attribute table is modified according to the previous description.

In Arc Info, to get rid of the second problem (cells in the no data field which have now the value -9999), the *reselect* command is used to keep only the cells with a positive value.

Arc: **reselect wshddiss wshddiss1**
Reselecting POLYGON features from WSHDDISS to create WSHDDISS1
Enter a logical expression. (Enter a blank line when finished)
>: **res grid-code > 0**
>:
Do you wish to re-enter expression (Y/N)? **n**
Do you wish to enter another expression (Y/N)? **n**

The resulting coverage is then dissolved to eliminate the first problem.

Arc: **dissolve wshddiss1 wshddiss2 grid-code**

The final coverage wshddiss2 has the same number of polygons as the grid.

The digital elevation model has been processed and the flowaccumulation and flowdirection grids on which the model is based have been created. Watersheds and creeks have been delineated and compared with observed data.

Chapter 4: Input parameters

The load is equal to the product of the discharge and the concentration, which both depend on precipitation and on land use. These two elements are the input parameters of the model (Figure 4.1).

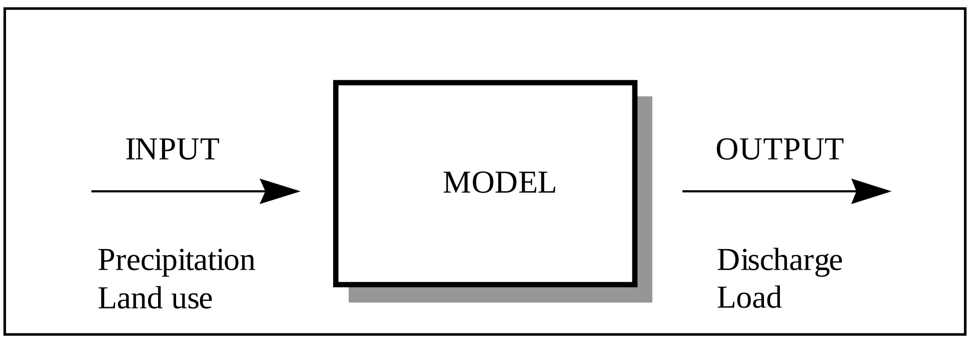


Figure 4.1: Model input and output

4.1 PRECIPITATION

The annual average rainfall given by the City of Austin to use for loading computation is 31.08 inches. It was computed according to EPA procedures using hourly data collected at the Austin airport during the period 1948-1993. Any "event" that did not generate at least 0.05 inches of rainfall within 6 hours was deleted. The mean annual storm event is one that occurs, on average, 51.8 times a year, has a volume of 0.60 inches, a duration of 7.8 hours, an average intensity of 0.106 inches/hour, and with 172.1 hours between storm event midpoints. In Austin, it rains about every 7 days on average.

Annual precipitation data may be obtained from the National Climatic Data Center (NCDC) Website (*http://www.ncdc.noaa.gov/coop-precip.html)*. There are three stations in Austin, but the Airport site is the only one with a continuous record since 1948. The average annual value for this period is 32.00 inches. For the period 1985-1994, the annual average rainfall volume is 34.37 inches. The value used (31.08 inches) underestimates slightly the volume of precipitation. However, in comparison with the variations in annual precipitation from year to year (from 12 to 55 inches), the difference is negligible (Figure 4.2).

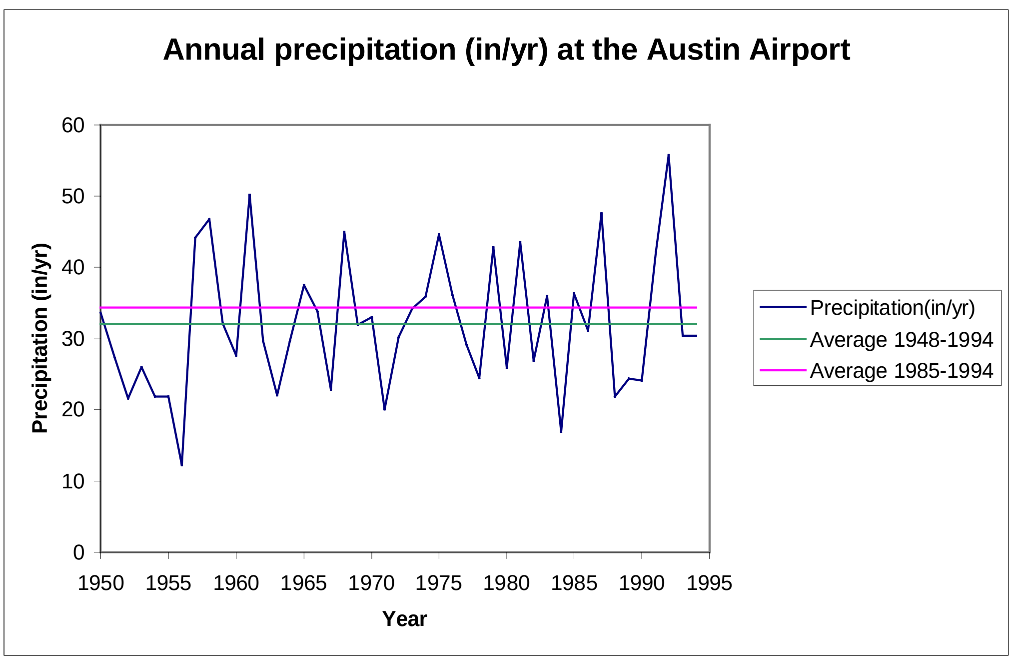


Figure 4.2: Annual precipitation at the Austin airport

4.2 LAND USE

The discharge depends on the part of the rainfall which contributes to the flow. This percentage is assumed to be related to the land use through relationships with the impervious cover.

4.2.1 Current land use

- **Impervious cover/Land use relationships**

The City of Austin has chosen to classify land use into nine categories (plus water) for its Water Quality Master Planning project (Table 4.1).

Table 4.1: Land use categories for the study

Single Family	Office	Park
Multi Family	Industrial	Transportation
Commercial	Civic	Undeveloped

The City has assigned to each category a range of impervious cover shown in Table 4.1, which is used to determine the runoff coefficients and the concentrations of the constituents (EMCs). A distinction was made between urban and non urban areas (Figure 4.3). The urban areas correspond to the city chore where each land use is assumed to have a higher impervious cover. The urban areas were assigned the upper limit of the range and the non urban areas the lower limit. The urban areas include the following watersheds (Table 4.2).

Table 4 2: Urban watersheds

Blunn	Town Lake	Buttermilk
East Bouldin	Johnson	Little Walnut
West Bouldin	Shoal	Fort Branch
Harper's Branch	Waller	Tannehill
Country Club	Boggy	

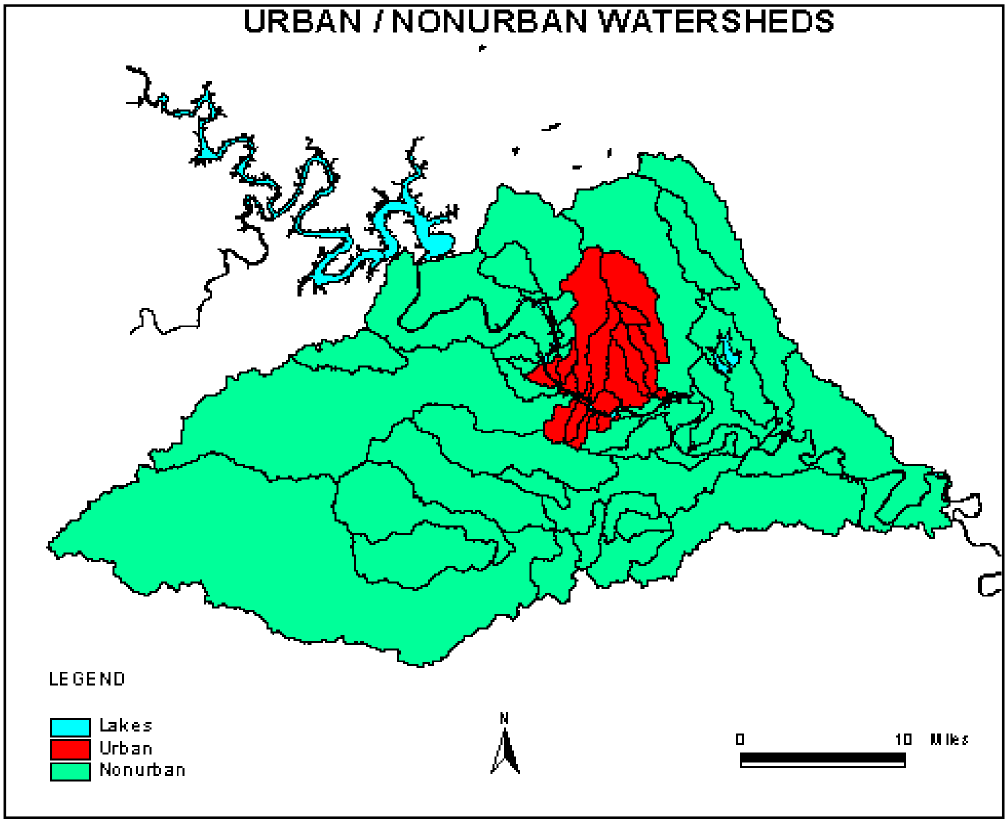


Figure 4.3: Urban and Non urban watersheds

The relationship between the land use and the impervious cover are shown in Table 4.3 for the land use categories considered.

Table 4.3: Land use-Impervious cover relationship

Land use		Impervious cover (%)	
Category	Code	Urban	Non urban
Single family	100, 113	40	30
Multi family	200	80	45
Commercial	300	95	60
Office	400	95	60
Industrial	500, 560	95	60
Civic	600	70	30
Park	700	15	5
Transportation	800, 870	100	85
Undeveloped	900, 999	15	5
Water	940	100	100

Most streets and roads are included in the overall land use they support. The transportation land use is reserved for only the largest of roadways (IH-35, MoPac) or for other uses such as railroads, but not for average city streets.

- **Land use coverage**

The Planning and Development Department of the City of Austin supplied the land use coverage used in the model. It contains some extra categories, which have to be reclassified among the existing ones (Table 4.4).

Table 4.4: Extra City of Austin land uses categories

Coverage		New classification	
Code	Category	Code	Category
113	Mobile home	100	Single family
560	Open extraction	500	Industrial
870	Utilities	800	Transportation
999	Unknown	900	Undeveloped

The category “utilities” was defined as equivalent to transportation.

The land use coverage supplied by the City (*landuse*) does not cover the entire study area and must be supplemented with the USGS land use coverage which can be downloaded from the EPA (*EPAluse*, http://www.tnris.state.tx.us/ftparea.html). The first step is to keep only the part of *EPAluse* which is inside the study area defined by the polygon coverage *border*. This new coverage is then used to complete the missing part. Those coverages come from two different sources. The city land use was updated in 1990 whereas the USGS land use was created in the 1960's. However, as the missing parts are located in mostly undeveloped areas, the land use has little changed during the interval.

Arc: **intersect EPAluse border luseborder**
Arc: **union landuse luseborder finluse**

The resulting polygon coverage *finluse* contains the fields from the two input coverages. The objective is to obtain a coverage following the City of Austin classification for the entire study region. The principle is to create a new land use code column in the attribute table (*newlandusecode*). The values in this field are equal to the City codes when applicable. However, USGS data used to complete the land use coverage employ a different code, which has first to be converted to the City standard (Table 4.5).

Table 4.5: USGS/City of Austin land uses classifications

USGS		City of Austin	
Code	Category	Code	Category
11	Residential	100	Single family
12	Commercial services	300	Commercial
13	Industrial	500	Industry
14	Transportation	800	Transportation
16	Mixed urban or build-up land	600	Civic
17	Other urban or build-up land	600	Civic
21-43	Cropland, pasture, orchards, groves, vineyards, agriculture, forest	900	Undeveloped
53	Reservoir	940	Water
75	Strip mines, quarries	900	Undeveloped
77	Mixed barren lands	900	Undeveloped

The edited *newlandusecode* field is used to dissolve the coverage in Arc/Info so that the resulting coverage represents the different land use polygons according to the City of Austin classification (Figure 4.4).

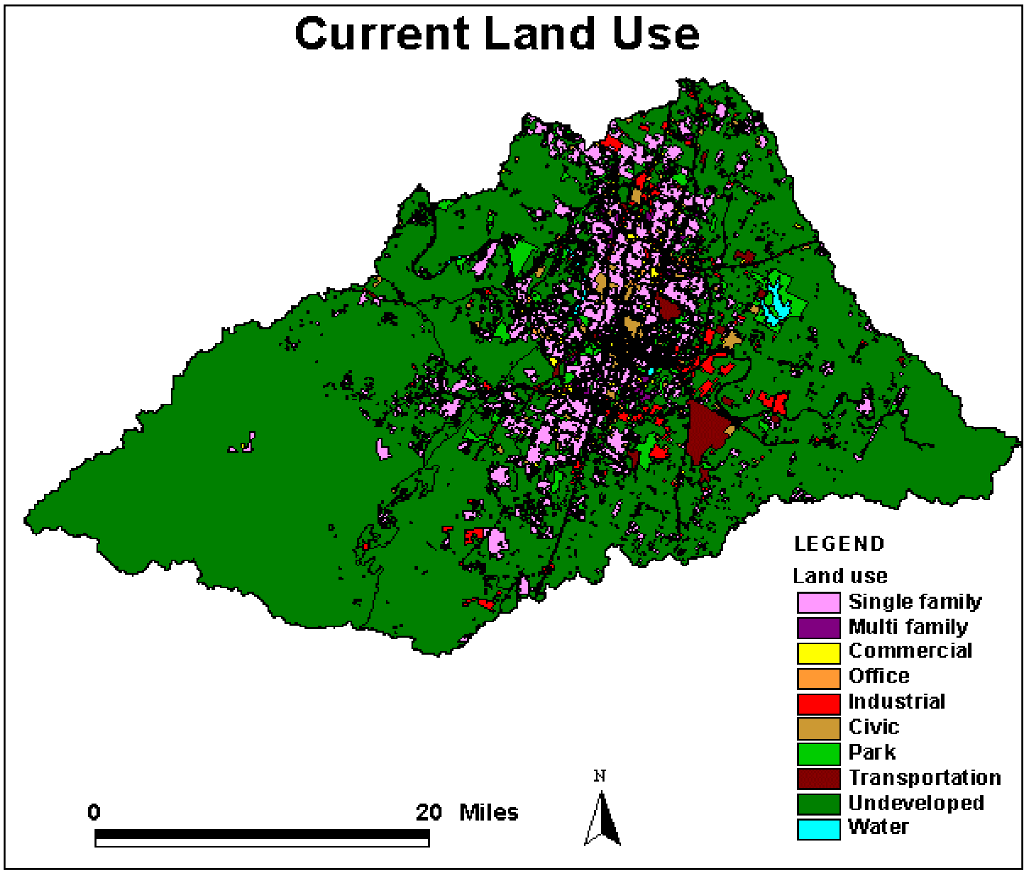


Figure 4.4: Current land use

The impervious cover can then be directly obtained (Figure 4.5) by applying the land use/impervious cover relationships to create the corresponding *IC* fields in the coverage attribute table.

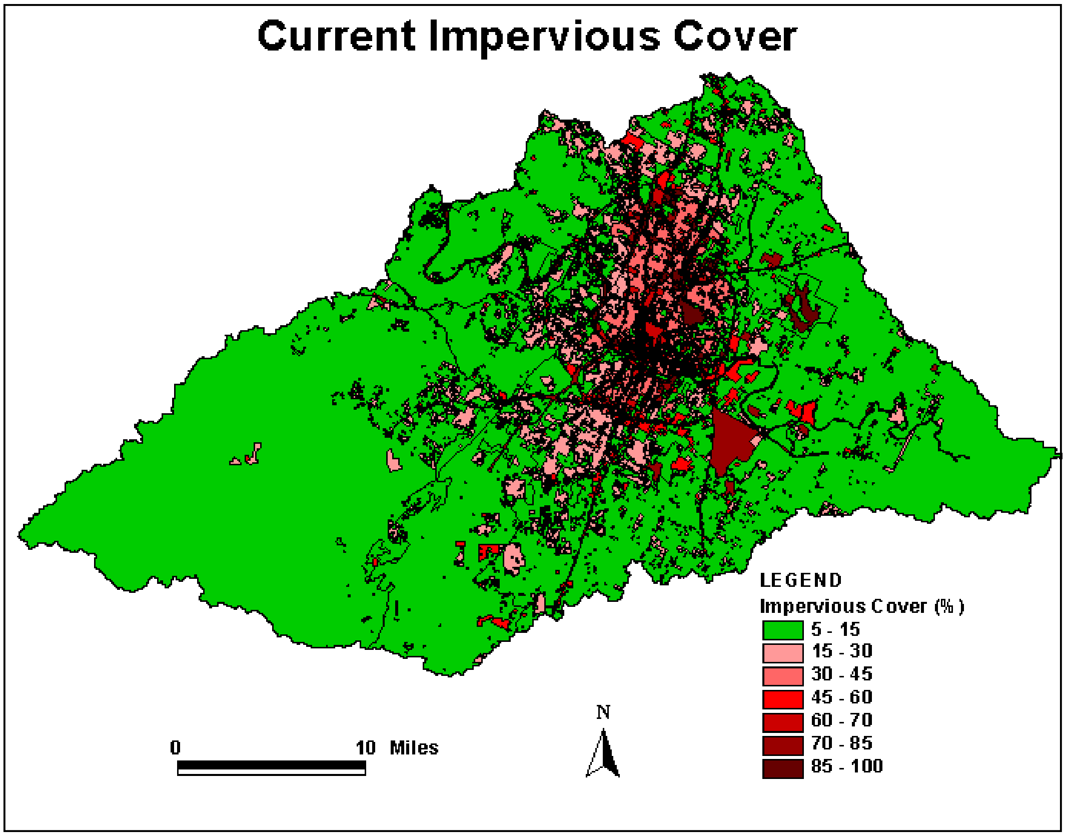


Figure 4.5: Current impervious cover

4.2.2 Future land use

- **Traffic Serial Zones**

It is difficult to accurately forecast the future distribution of land uses, so there are represented as an average within the traffic serial zones. The traffic serial zones are areas used in transportation studies: each zone is delimited by roads with a given traffic. The predicted future impervious cover was computed by the City of Austin. Starting from the current land use within the different zones, the City of Austin made some assumptions about the future development. The impervious cover at any location can only increase. Figure 4.6 shows the increase between future and current impervious cover.

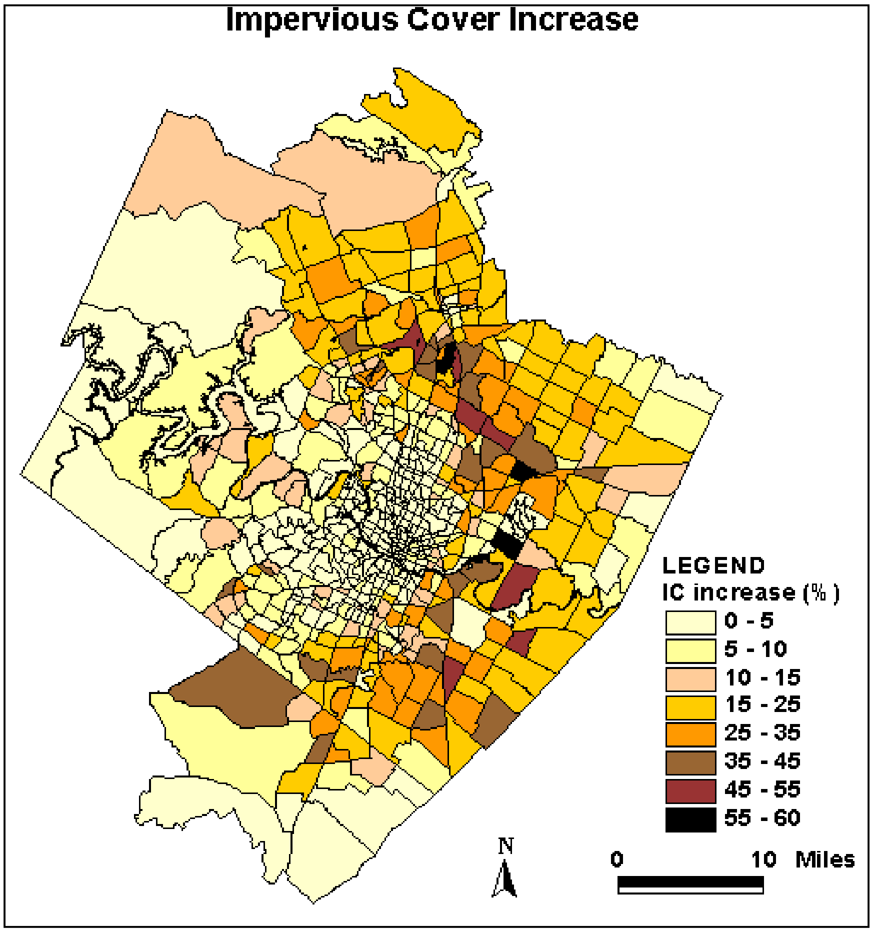


Figure 4.6: Impervious cover increase in the traffic serial zones

To compare current and future impervious cover, the current values must first be averaged over the traffic serial zones. The average value in a zone is computed with the *zonalmean* function in Arc/Info Grid.

Grid: **zones_gr = polygrid (zones)**
Grid: **ic_gr = polygrid (finluse , ic)**
Grid: **iczone_avg = zonalmean (zones , ic_gr)**

A script (*Qual.Zonalmean*, Appendix C) was written so that the zonal average can also be computed in ArcView. The script was customized with the button . Two message boxes prompt the user for a zone coverage and for a value theme (grid or coverage).

The comparison between current and future average impervious covers in the traffic serial zones shows that the impervious cover decreases in a few zones. The problem arises because the current impervious cover is defined in two different ways. To guarantee an increase in impervious cover, the future impervious cover in the traffic serial zones must be defined as the higher value of the current and future average values. This can be done in Arc/Info by taking the difference between the future and current average impervious cover grid, and by replacing the future impervious cover by the current one in the future impervious grid in the zone where the difference of impervious cover is negative.

Since the traffic serial zones do not cover the whole area, the missing parts have to be completed. There are two main missing zones. The first consists in the western parts of Barton, Onion and Bear watersheds while the second corresponds to the part of Dry and Colorado watersheds. As Barton belongs to the study area, it is important to complete the zones with an accurate value. It is assumed that this zone has the same development as the traffic serial zone 43 which is located in Barton (Figure 4.7). For the other zone, two values are defined corresponding to the average values of the closest traffic serial zones located in the same watershed. However, the definition in that area is not very important because they do not belong to the watersheds studied and because they are at the limit of the study (downstream part of the watershed is missing). In Arc/Info the function *union* enables one to create a complete zone coverage over the entire study area. The impervious cover corresponding to the new zones are added in the *TSZ* field which identifies the different zones.

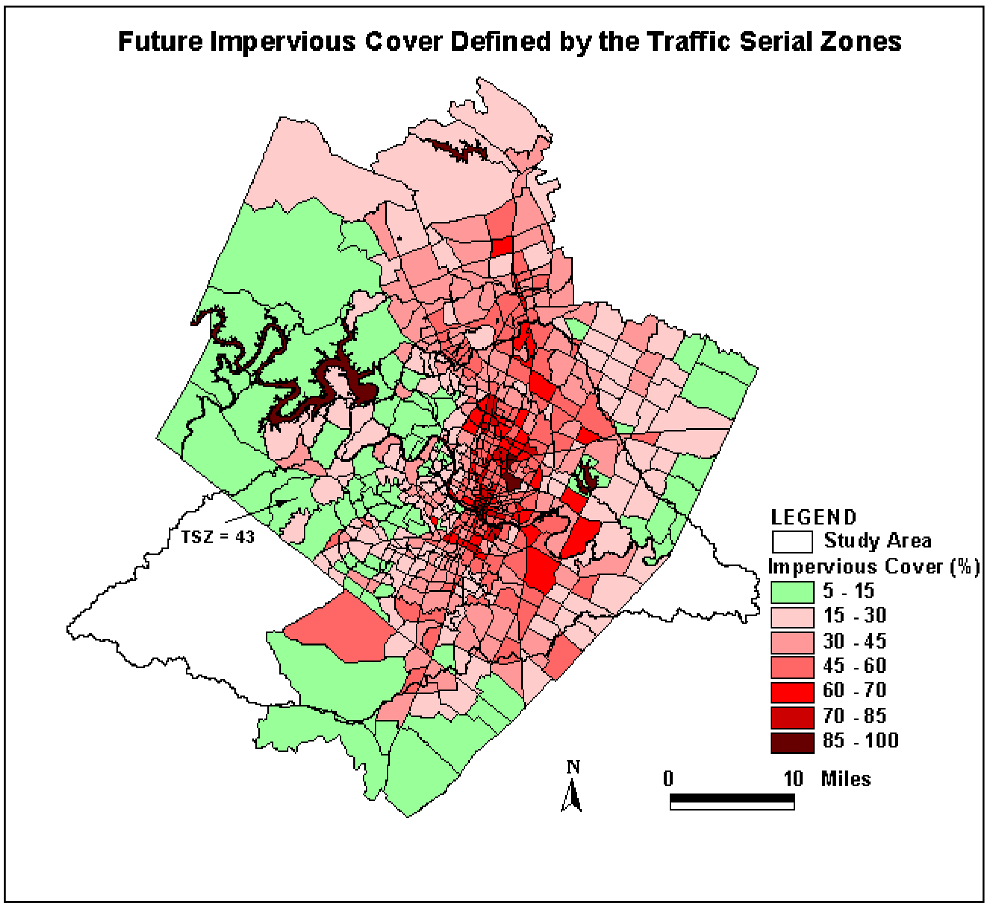


Figure 4.7: Future impervious cover defined by the traffic serial zones

- **Comparison between current and future impervious cover**

The average impervious cover value at any location within the area of study can be obtained by using the function *flowaccumulation*. The script *Qual.Average* (Appendix C) was written so that the average value could be directly computed in ArcView. This script is customized with the Σ button and the command *QualTools/Average*. The program is based on the principle used in Arc/Info. The required input data are a flowdirection grid, a flowaccumulation grid and either a coverage or a grid containing the value to average (in this case, the impervious cover).

The comparison between the resulting current and future average impervious covers shows that some problems still remain. The current impervious cover is higher than the future one for some locations. This is due to the fact that the average definition smoothes the impervious cover values. For some locations, like in Jo [4.1] d where the current impervious cover is equal to 100% (urban transportation), the values for points located in the upstream part of the watersheds can be very high for current conditions and lower for future which are averaged. To guarantee that the impervious cover at any location increases, the problem caused by the difference of accuracy in the definition of the data has to be solved. The alternatives are to either decrease the current accuracy or increase the future one. To keep the accuracy of the current coverage, the future impervious cover is defined from the current one so that the impervious cover is increasing in each cell (Figure 4.8). The method, presented by Dr. Francisco Olivera, consists in defining the future impervious cover as the sum of the current one and of a positive number depending on the average impervious cover value associated with the traffic serial zone and on the current impervious cover in the cell.

$$IC_F = IC_C + (1 - IC_C) * \frac{\overline{IC_F} - \overline{IC_C}}{1 - \overline{IC_C}}$$

Where $\overline{IC_F}$ *and* $\overline{IC_C}$ are respectively the average future and current impervious covers over each traffic serial zone.

This relationship is set up so that:

- $IC_F > IC_C$
- $IC_F < 1$
- The average value of IC_F over the TSZ ($\overline{IC_F}$) is the same as the value associated to the corresponding TSZ.
- Less developed areas tend to develop more than already developed areas.

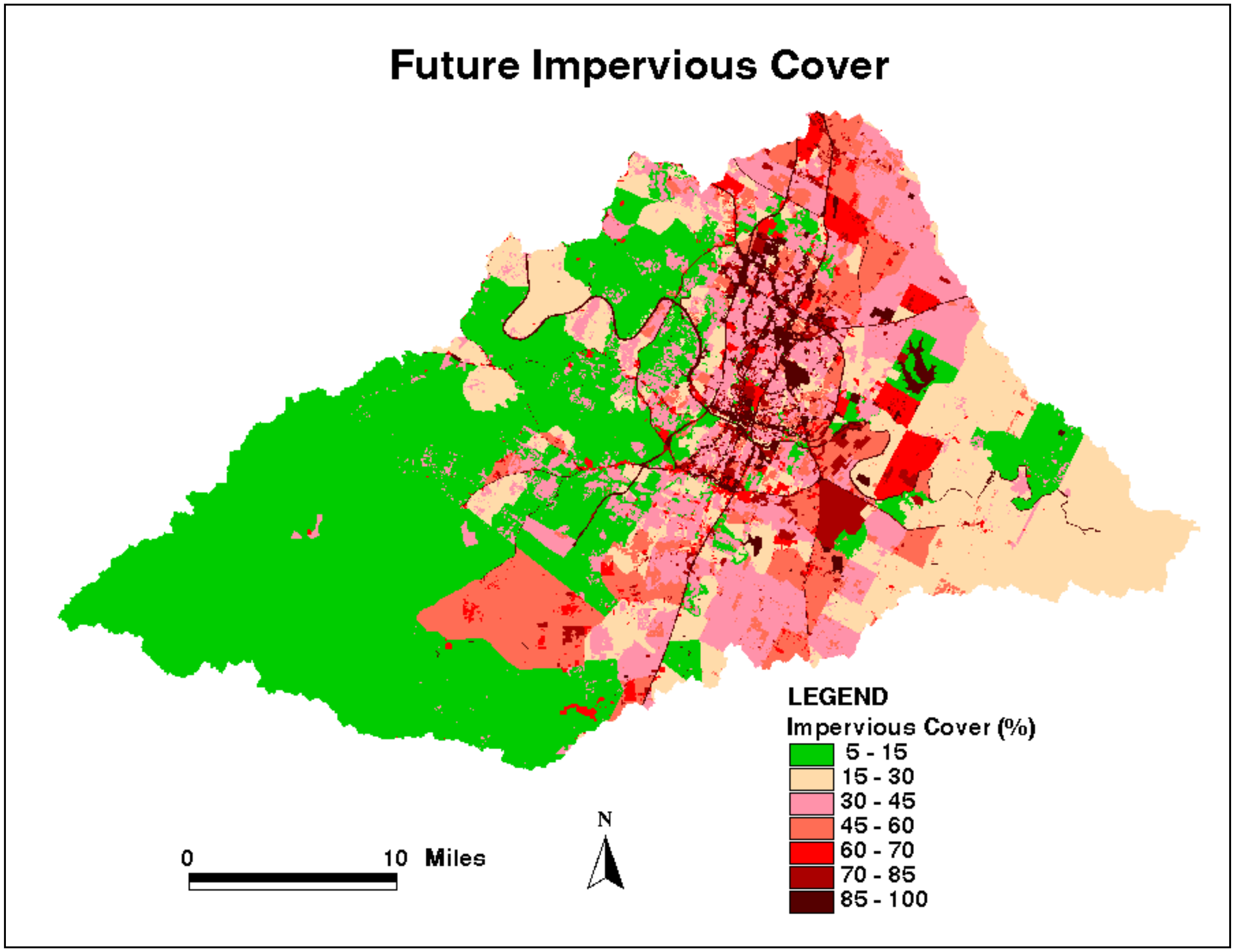


Figure 4.8: Future Impervious Cover

The input parameters (precipitation, land use) used in the model have been defined, as well as the relationships between land use and impervious cover. A procedure to relate future and current impervious cover has been implemented so that impervious cover increases with time.

The input parameters are used with the flowdirection and flowaccumulation grids previously defined to compute flow, load and BMP effects.

Chapter 5: Discharge

5.1 COMPUTATION METHOD

Several factors were considered in the computation of the discharge. Direct runoff and base flow components of flows were initially calculated, not accounting for flow losses in the recharge zone. Then, losses were calculated (in base flow and runoff components) and subtracted from flows occurring in the recharge zone. Finally, predicted flows were corrected to match the flows observed at the USGS stations.

5.1.1 Two modes of contribution

Rainfall contributes to stream discharge in two ways (Figure 5.1):

- by direct (surface) runoff.
- by infiltration and contribution as base flow.

The discharges computed are annual average data, therefore the lag time between rainfall and flow does not have to be considered, which greatly simplifies the model since this lag time is different for direct runoff and for base flow. Two runoff coefficients determine the percentage of precipitation volume contributing to the discharge as direct runoff (Rv) or base flow (Rv_{bf}) (Figure 5.1).

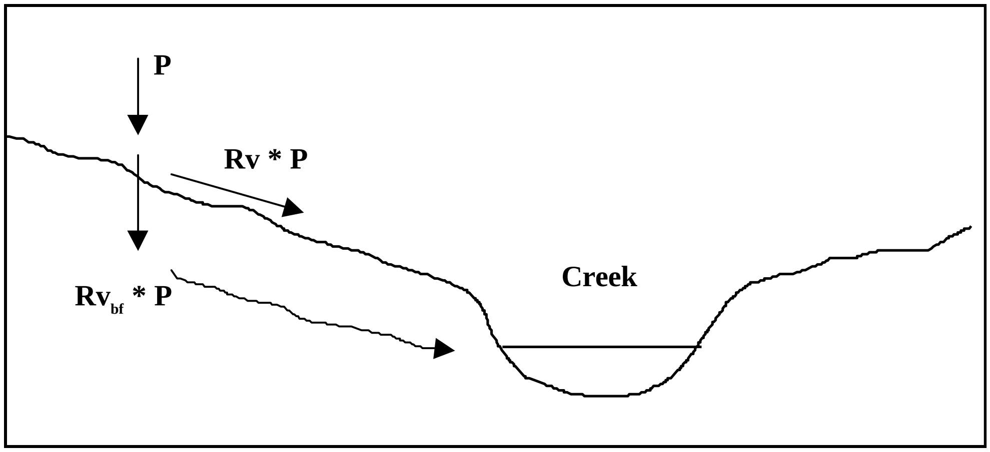


Figure 5.1: The two modes of contribution of precipitation

These coefficients are a function of the impervious cover, which is a function of land use. Relationships between impervious cover (IC) and runoff coefficients (Rv and Rv_{bf}) were calculated by Dr. Michael Barrett of CRWR (Barrett, 197) from observed data (Figures 5.2 and 5.3).

$$\mathbf{Rv = 0.3428 * IC^2 + 0.5677 * IC + 0.0125, \text{ with } 0 < IC < 1} \quad [5.1.a]$$

$$\mathbf{Rv_{bf} = -0.36 * IC + 0.1904, \text{ with } 0 < IC < 1} \quad [5.1.b]$$

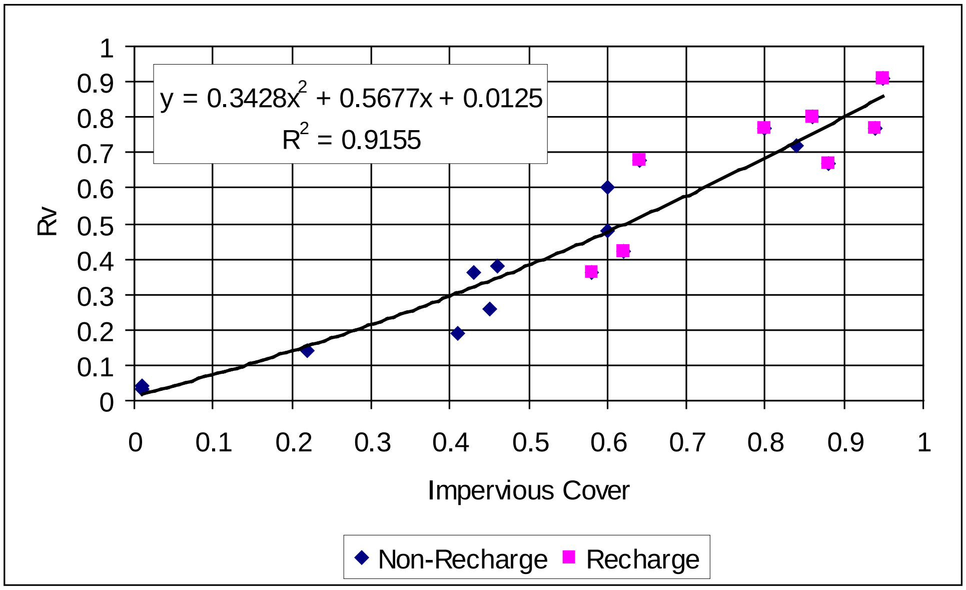


Figure 5.2: Relationship between direct runoff coefficient and impervious cover

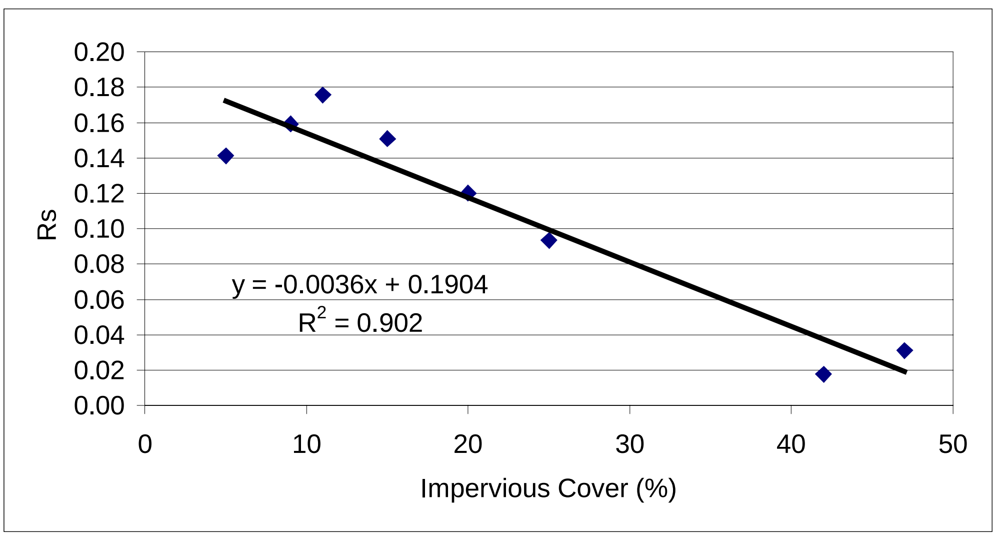


Figure 5.3: Relationship between base flow coefficient and impervious cover

The relationship used to compute the base flow coefficient yields negative values for impervious cover values bigger than 52%. Since it must be a positive number, the base flow coefficient is then set to zero: no base flow occurs when the impervious cover is bigger than 52%.

The total amount of rainfall received by a water body contributes to the discharge, which corresponds to a runoff coefficient of 1. However, the impervious cover associated with water as a land use (100%) yields a computed total runoff coefficient of 0.923, which was used in the study (evaporation from water body).

5.1.2 Discharge produced by each cell

The annual precipitation of 31.08 inches is applied uniformly over the region. The contribution of any given cell to the discharge (in cfs) can be obtained by multiplying the runoff coefficient by the annual average volume of precipitation and the necessary units conversion coefficients.

The direct runoff (Figure 5.4) is found as:

$$Q_r\,(cfs) = \frac{P(in/\,year) * Rv * Cell\ \text{area}(\,ft^2)}{86400(s/\,day) * 365(days/\,year) * 12(in/\ ft)}$$

$$= \frac{P(in/\,year) * Rv * 10000(\,ft^2)}{86400 * 365 * 12}$$

$$= \frac{P(in/\,year) * Rv * 5}{189216} \qquad [5.2]$$

and similarly the base flow as:

$$Q_{bf}\,(cfs) = \frac{P(in/year) * Rv_{bf} * 5}{189216} \qquad [5.3]$$

to produce the total discharge

$$Q_t\,(cfs) = Q_r + Q_{bf} \qquad [5.4]$$

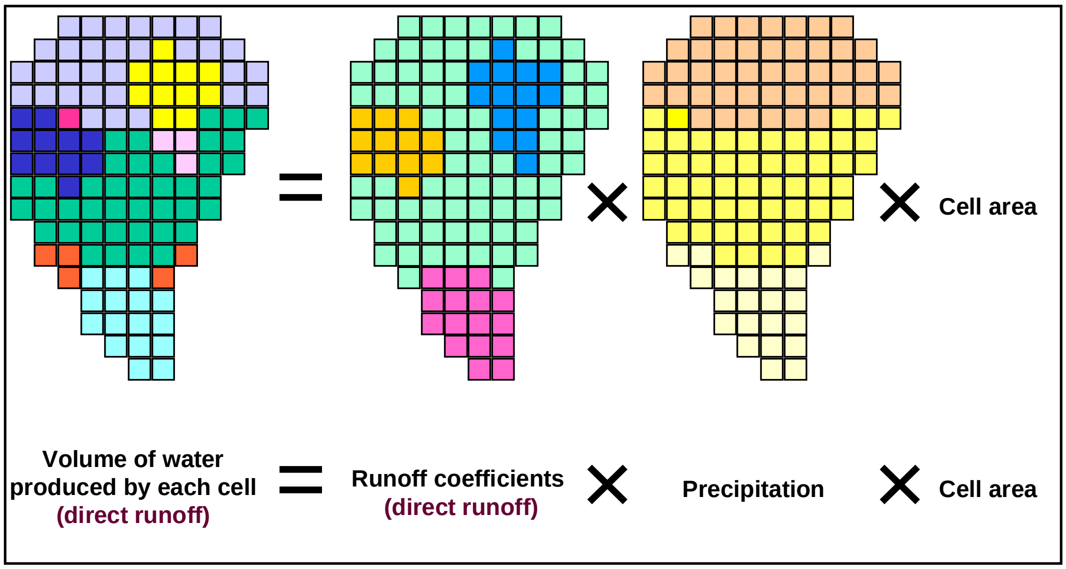


Figure 5.4: Direct runoff produced by each cell

5.1.3 Total discharge

The total discharge in any cell is equal to the sum of the contributions of all the cells located upstream of that cell, which is calculated using the Grid *flowaccumulation* function with discharge as the weight grid:

$$Q_{cell} = \sum_{\substack{upstream \\ cells}} Q_t = Flowaccumulation(Fdirgrid, Q_t\,grid) \qquad [5.5]$$

where Fdirgrid is the flowdirection grid and $Q_t$grid contains the calculated discharges contributions from each cell.

5.1.4 Discharge computation

Discharge can be computed directly from the land use coverage by using a program in Arc Info. The AML used in the computation allows one to compute the direct runoff, the base flow and the total flow (Procedure 5.1). The fields corresponding to the runoff coefficients (*runcoef*, *runcoef_bf*) are created in the attribute table of the land use coverage (*fin_luse*).

Procedure 5.1: Discharge computation

```
/*FUNCTION: compute the base flow, runoff and the total discharge.
/*INPUTS: land use coverage (fin_luse) with fields for the runoff coefficients (runcoef
and runcoef_bf), flowdirection grid (burn_fdr), precipitation = 31.08 in/yr.
/*OUTPUTS: direct runoff grid (runoff), base flow grid (baseflow) and total flow grid
(rawflow) in cfs.

/*-----------------------------------------------
/*BEGIN AML EXECUTION
grid
setcell 100

/*Compute the runoff coefficient grids
runcoef_gr = polygrid ( fin_luse , runcoef )
bfruncoef_gr = polygrid ( fin_luse , runcoef_bf )

/*Compute the direct runoff and base flow produced by each cell
runoff_gr = runcoef_gr * 31.08 * 5 / 189216
baseflow_gr = bfruncoef_gr * 31.08 * 5 / 189216
kill runcoef_gr all
kill bfruncoef_gr all
```

```
/*Compute the total direct runoff, base flow and flow
runoff = flowaccumulation ( burn_fdr , runoff_gr )
kill runoff_gr all
baseflow = flowaccumulation ( burn_fdr, baseflow_gr )
kill baseflow_gr all
rawflow = runoff + baseflow
&return

/*--------------- END OF AML ------------
```

The three output grids, *rawflow*, *runoff* and *baseflow*, contain respectively the values of the total discharge, direct runoff and base flow for any cell in the grid. The values at selected locations (e.g. USGS stations in table 5.1) can be obtained by using the script *Qual.Pick* in ArcView (section 2.2.3).

Table 5.1: Base flow, direct runoff and total flow at the USGS stations based on impervious cover/runoff coefficient relationships

USGS #	Name	Stormflow (cfs)	Baseflow (cfs)	Totalflow (cfs)
8155300	Barton at Loop 360	15.755	44.162	59.917
8155240	Barton at Lost Creek	12.785	41.103	53.888
8155200	Barton at SH 71	9.887	34.652	44.539
8158810	Bear at FM Road 1826	1.538	4.65	6.188
8158050	Boggy at US183	13.173	1.42	14.593
8154700	Bull at Loop 360	5.746	7.288	13.034
8158800	Onion at Buda	17.353	65.366	82.719
8158700	Onion at Driftwood	12.195	48.545	60.74
8159000	Onion at US 183	52.522	117.966	170.488
8156700	Shoal at NW Park	7.194	0.571	7.765
8156800	Shoal at W 12th	13.373	1.064	14.436
8158840	Slaughter at FM Road 1826	2.02	2.936	4.956
8157500	Waller at 23rd	4.997	0.27	5.268
8157000	Waller at 38th	2.668	0.178	2.846
8158600	Walnut at Webberville Road	28.68	12.955	41.635
8158970	Williamson at Jimmy Clay Road	10.073	7.466	17.539
8158920	Williamson at Oak Hill	1.783	1.948	3.731

5.2 RECHARGE

5.2.1 Difference between recharge and non recharge zones

Flows are now calculated for the entire study area. However, in-stream losses for portions of the watersheds that are in the recharge zone of the Edwards Aquifer must be taken into account. The losses occur only in the recharge zone. However, flow computations are conducted at the same manner for watersheds located inside and outside of the recharge zone, by saying that no losses occur outside of the recharge zone. However, considering losses (even zero losses) increases the number of operations in the program, it is better to define a simplified program in the case where no recharge zone has to be considered. Everything described in section 5.2 is then irrelevant and flows defined by the Procedure 5.1 are used in further computations (section 5.4) as the flows obtained after losses. The analysis was originally done in Arc/Info and later transferred to ArcView and programmed in Avenue for ease of use by the City of Austin. The final Arc/Info program does not consider the no recharge case, since it aims at explaining the general, more complicated, methodology. However, the Avenue script for flow computation offers the user the option not to consider a recharge zone (section 8.2).

5.2.2 Recharge losses coefficients

Portions of the watersheds in this study overlay either the contributing or the recharge zone of the Edwards Aquifer (only the southern zone is considered since no data are available for the northern zone). A study of the Edwards aquifer (Barrett and Charbeneau, 1996) has determined the annual average aquifer recharge values for the different creeks flowing through the recharge zone as shown in column 2 of Table 5.2.

Losses to the recharge zone are assumed to occur in the creeks and to be proportional to the channel length within the recharge zone. Figure 5.5 shows two of the watersheds which are partially in the recharge zone (Bear and Little Bear).

.

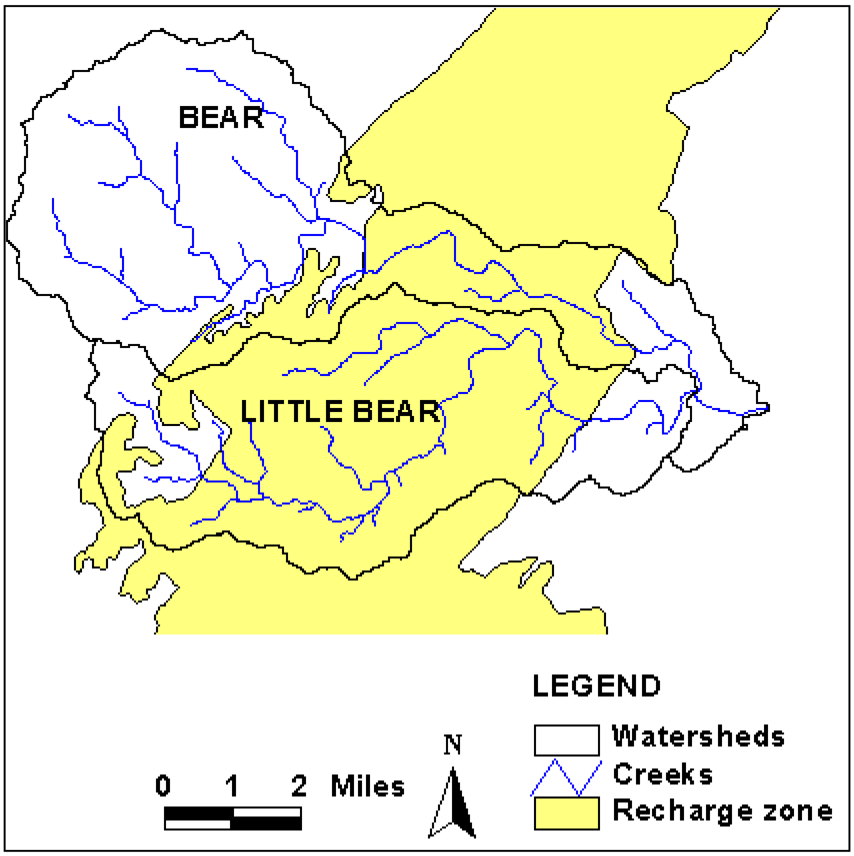


Figure 5.5: Example of watersheds in the recharge zone

Hence, the recharge at any given cell located in a creek within the recharge zone is obtained by dividing the total recharge occurring in that creek by the total length of the channel inside the recharge zone, and by multiplying by the length of the channel in that cell (Eq. 5.6).

$$\text{Recharge coefficient}\left[\frac{\text{cfs}}{\text{cell}}\right] = \frac{\text{Total creek recharge}}{\text{Length of creek in recharge zone}} \times \text{cell length} \quad [5.6]$$

- **Channel length in the recharge zone**

The channel length inside the recharge zone is defined by selecting the part of the main creek in the recharge zone. This is because all the recharge is assumed to occur only in the streams, and therefore is larger per cell than the runoff which is distributed over the area. For tributaries corresponding to small drainage areas, the total flow generated is less than the losses to the recharge, and the flow obtained after subtracting the losses is negative. Considering that losses occur only in the mainstems, the negative flows are eliminated in such tributaries.

The cell size is 100 feet. It is assumed that the flow can only go through the opposite sides or through the diagonal, thus the length will be either 100 ft or 100ft * √2, i.e. 141.4ft (Figure 5.6).

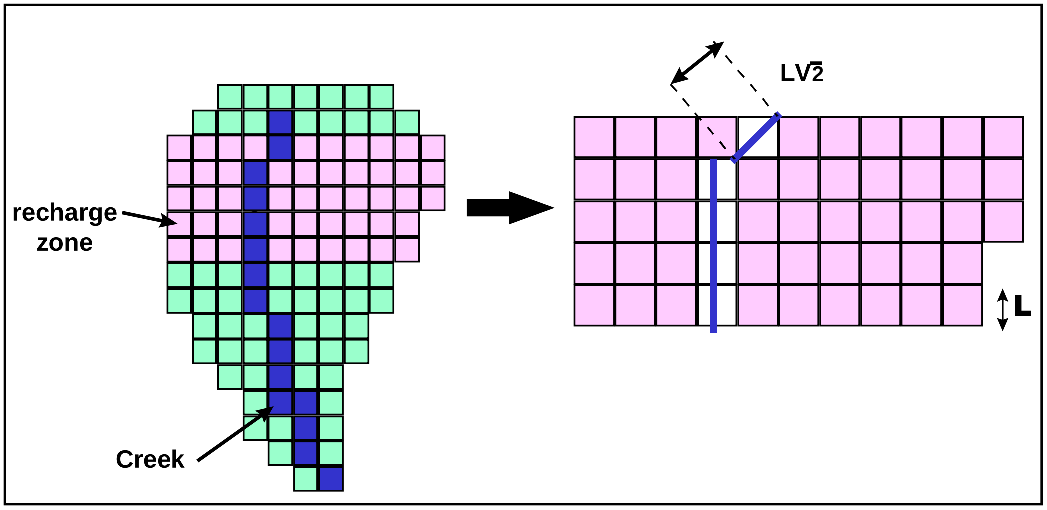


Figure 5.6: Flow length in the recharge zone

The creeks are defined by choosing an appropriate threshold value (1,000 cells) which enables, for the most part, selection of only the main streams which closely correspond to the actual GIS stream coverages provided by the City. The portion of the stream network *crk1k_gr* (section 3.4) within the recharge zone is then selected by using the function *selectpolygon* in Arc/Info.

Grid: **lrech_gr = selectpolygon (crk1k_gr , recharge)**

The grid is corrected in Arctools so that only the main streams remain, and then merged with a grid whose cells have for value 0 (0_gr). The zero valued cells indicate areas that are not streams.

Grid: **0_gr = con (burn_fdr , 0)**
Grid: **lrech1 = merge (lrech_gr , 0_gr)**

Each creek cell is assigned a value corresponding to the length of the channel in that cell: the value is either 100ft (longitudinal) or 141.4ft (diagonal). That length is determined by using the flowdirection grid which codes the flow direction (100ft for value 1, 4, 16, 64 and 141.4ft for 2, 8, 32, and 128, see Figure 3.4).

Grid: **lrech2 = con(crk1k_gr == 1 and (burn_fdr == 1 or burn_fdr == 4 or burn_fdr == 16 or burn_fdr == 64) , 100 , 141.4)**

The length grid (*lrech2*) is then multiplied by the grid with the main streams (*lrech1*) in the recharge zone (whose cell value is 1) to create a grid containing main creek cells in the recharge zone with channel length (*lrech*).

Grid: **lrech = lrech1 * lrech2**

Channel length for creeks in the recharge zone is obtained by adding the length value of each cell for each creek. The sum of the cell lengths within each watershed is computed with the function *zonalsum* in Grid.

Grid: **lcount = zonalsum (coawshd_gr , lrech)**

This results in a watershed grid whose grid-code is the total channel length of each creek in the recharge zone. Note that the recharge value for Bear includes the recharge value for Bear Creek and Little Bear Creek. Hence the total channel length corresponding to the discharge value for Bear must be the sum of the channel length computed for Bear and for Little Bear.

- **Recharge coefficients**

The recharge coefficients applied to each watershed are obtained by dividing the recharge value by the corresponding channel length (Eq. 5.6). The recharge coefficients are computed in ArcView and written to the field *rechcoef* (Table 5.2) in the attributes table of the watershed coverage (*coawshd_cv*).

Table 5.2: Recharge coefficients

Watershed	Recharge (cfs)	Rech_length (ft)	Rechcoef (cfs/ft)	Newrechcoef ** (cfs/ft)
Barton	20	37134	5.39E-04	5386
Bear*	9	83758	1.08E-04	1075
Onion	31	58156	5.33E-04	5330
Slaughter	3.5	61333	5.71E-05	571
Williamson	1.9	57315	3.32E-05	332

* including Little Bear

** newrechcoef = rechcoef * 10^7

The recharge coefficient grid is then created from the coverage. The grid resulting when using the recharge coefficients, which are very small (10^{-4}), as grid-code is a zero valued grid. To avoid this problem, the field *newrechcoef* containing the recharge coefficients multiplied by 10^7 are used instead to create the grid.

Grid: **rechcorr_gr = polygrid (coawshd_cv , newrechcoef)**

The recharge flow produced by each cell is obtained by multiplying the correction coefficient grid (*rechcorr_gr*) by the recharge channel length grid (*lrech*), and then dividing by 10^7.

Grid: **lcorr_rech = lrech * rechcorr_gr / 10000000**

An ArcView script allowing the user to create the grid *lcorr_rech* is currently developed at CRWR and should be available by December 15, 1997.

5.2.3 Calculating flows with recharge losses

The *flowaccumulation* program does not handle negative cell values. Since the recharge occurs only in the streams whereas the flow is generated over the area, the losses to the recharge zone are bigger than the flow generated, and the flow generated when considering the losses is negative. Therefore, the total flow in each cell is calculated first over the entire study area without considering recharge effects. Then, the recharge is subtracted.

The total recharge flow at any cell is obtained by summing the recharge for each cell by using the *flowaccumulation* function.

Grid: **rech_fac = flowaccumulation (burn_fdr , lcorr_rech)**

For each cell, the total flow (*flow0*) including the losses in the recharge zone is obtained by subtracting the recharge flow (*rech_fac*) to the flow computed in section 5.1.4 (*rawflow*).

Grid: **flow0 = rawflow – rech_fac**

The flow lost at a given location in the recharge zone is assumed to have the same ratio of base flow to total flow as the flow in that cell. It is then possible to determine the new values for direct runoff and base flow, as well as their respective volumes lost in the recharge zone. The ratio of base flow to total flow at any location is obtained by dividing the base flow by the total flow (without considering the effect of the recharge zone).

Grid: **partbflow = baseflow / rawflow**

This grid *partbflow* is then multiplied with *lcorr_rech*, which indicates the recharge flow produced in each cell, to yield the base flow contributing to the recharge produced in each cell. Summing over all the cells with the *flowaccumulation* function, a grid with the total recharge base flow at any cell is obtained. This grid is then subtracted from the base flow grid previously obtained (which did not consider recharge) to create the base flow grid (*newbaseflow*).

Grid: **rechcellbflow = partbflow * lcorr_rech**
Grid: **rechbflow = flowaccumulation (burn_fdr , rechcellbflow)**
Grid: **newbaseflow = baseflow – rechbflow**

5.3 FLOW CALIBRATION

The method presented in the two previous sections is used to compute predictive flows reflecting empirical relationships (Eq. 5.1.a and 5.1.b) and recharge zones assumptions. Calibration to observed values is necessary and important to ensure a realistic simulation for predictive purposes (Figure 5.7).

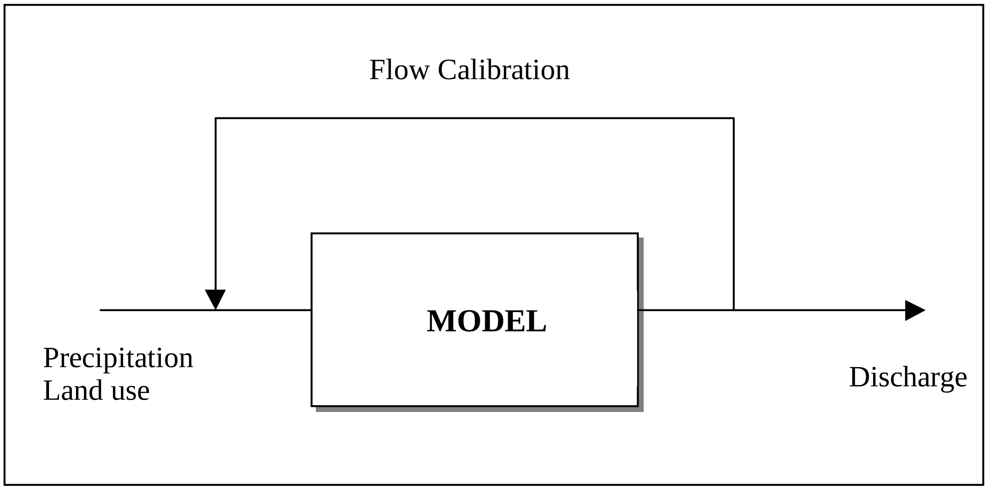


Figure 5.7: Flow calibration

5.3.1 Observed discharges

The available observed data is from the USGS sites. Approximately 30 gauging stations were located within the study area of which only 20 have a period of record that includes at least one complete year. Moreover, 17 stations have enough information to be used as references, but only 8 have values for each year of the period of record chosen (1985-1994), where the year begins on October 1 and ends on September 30 (water year). Water years were chosen over calendars years because there was more data available within a water year.

Missing annual average values were extrapolated from other data, with consideration given to the relative locations of the basins studied, their areas and their geological characteristics (e.g. recharge zone of the Edwards Aquifer).

The period of record was chosen based on a compromise between three factors. The objective is to be able to predict current and future variations in pollutant loads as a function of land use, which is a function of impervious cover. Therefore it is necessary to describe existing conditions as accurately as possible. A compromise must be reached so that:

- there are enough data
- the period is short enough so that there are no major modifications in land use.
- the period is long enough so that the discharge values are representative.

Section 5.3.2 describes a procedure for quantifying streamflow variability that was examined but could not be implemented, since probability characteristics can not be extrapolated, and is included in the study for documentation purposes.

5.3.2 Average discharge and probabilistic approach

The average annual value was used for precipitation and discharge. However, the period of record for the precipitation data is 1948-1993 and 1985-1994 for the discharge data. Nevertheless, the difference in the average values for both periods was not significant for precipitation (32.00 inches versus 34.37 inches for 1985-1994).

It is equally interesting to note the annual variations of the different discharges since they are directly related to the loads. Flow calibration is important because concentrations are fixed for a given impervious cover so that the load is directly proportional to the flow. For the period of record, the predicted loads are either over or underestimated. For example, the discharge at Barton Creek at Loop 360 varies from 0.14 cfs to 228 cfs (Figure 5.8). During the wet years, predicted pollutant load, using mean annual flow as input is largely underestimated whereas during the dry years, it is correspondingly overestimated.

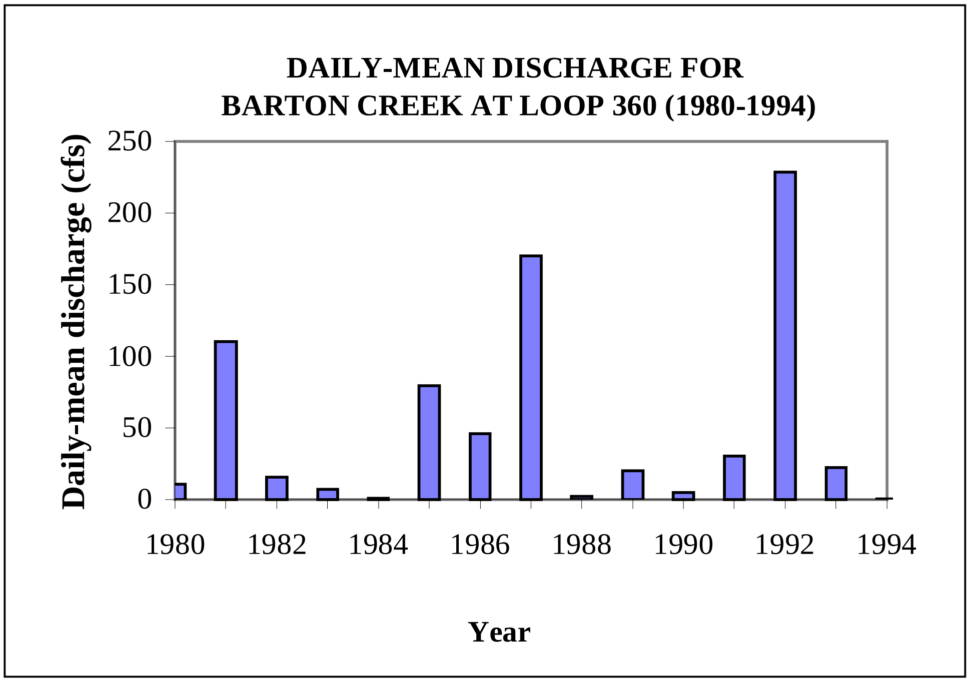


Figure 5.8: Daily mean discharge at Barton Creek at Loop 360 (1980-1994)

The use of a probabilistic approach, which would give a representation of the variations from the mean would be more realistic than considering only an average value. The study of the discharge data for a station with a complete record shows that a lognormal distribution fits the data well (Figures 5.9-5.12). Table 5.3 shows mean and standard deviation of $\log_{10}Q$ used in plotting the lognormal distribution figures.

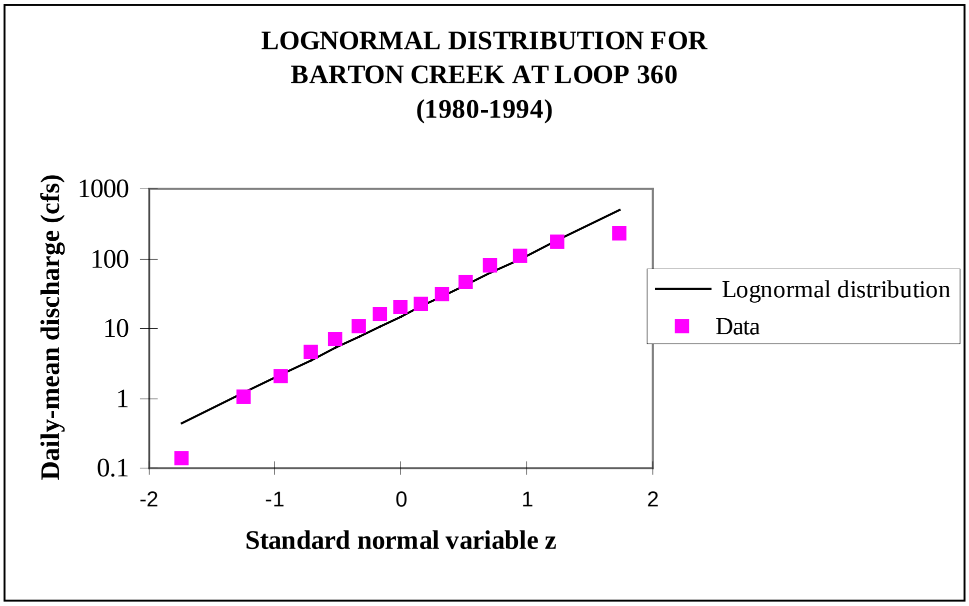


Figure 5.9: Lognormal distribution for Barton Creek at Loop 360

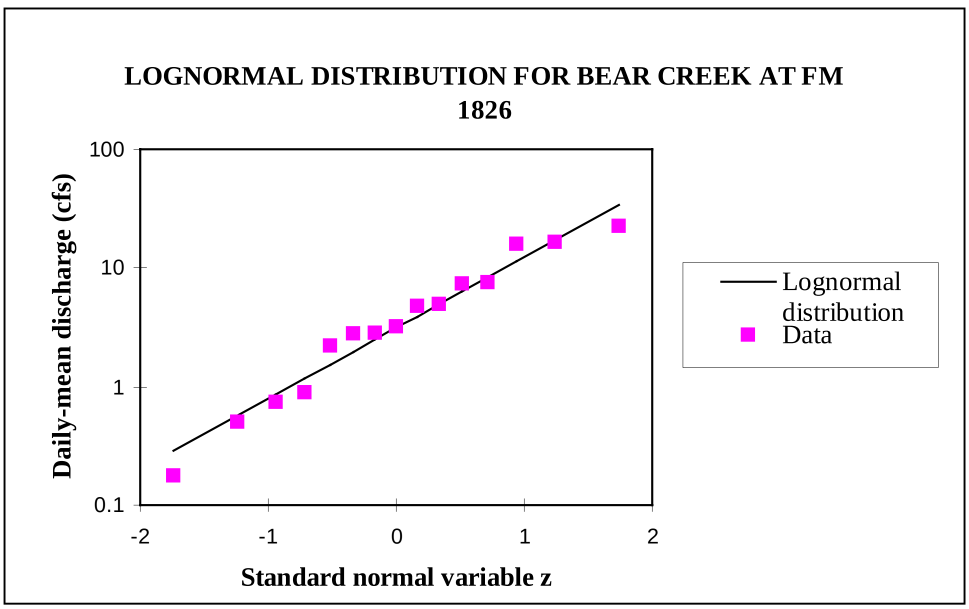


Figure 5.10: Lognormal distribution for Bear Creek at FM 1826

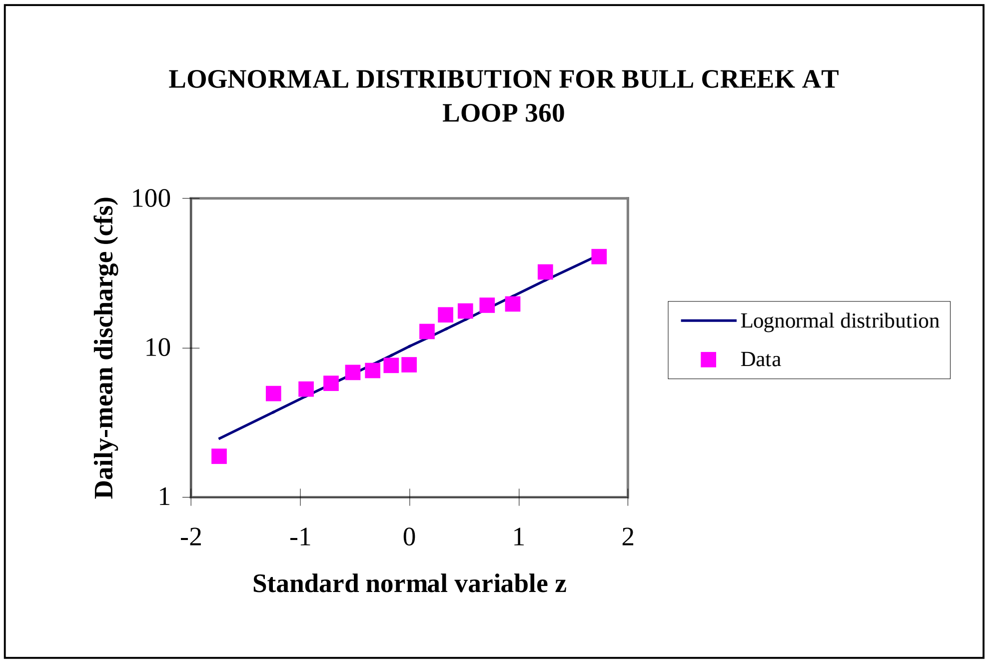


Figure 5.11: Lognormal distribution for Bull Creek at Loop 360

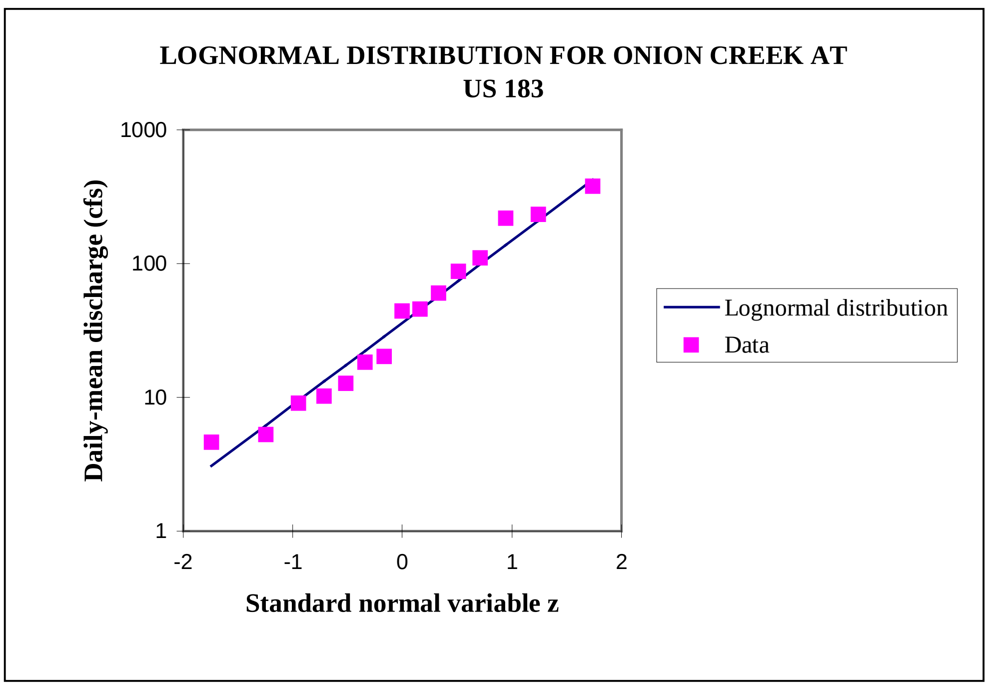


Figure 5.12: Lognormal distribution for Onion Creek at US 183

Table 5.3: Mean and standard deviation of $\log_{10}Q$

Station	Mean (μ)	Standard deviation (σ)
Barton Creek at Loop 360	2.86	2.24
Bear Creek at FM 1826	1.38	1.18
Bull Creek at Loop 360	2.54	0.76
Onion Creek at US 183	3.55	1.59
Onion near Driftwood	3.47	1.34
Slaughter Creek at FM 1826	0.59	2.17
Walnut Creek at Webberville road	3.34	0.69

The pollutant load for a given period is defined as the product of the concentration and the discharge:

$$\mathbf{LOAD = Q * C} \qquad [5.7]$$

Taking the natural logarithm of this expression:

$$\mathbf{Ln(Q*C) = Ln(Q) + Ln(C)} \qquad [5.8]$$

The concentrations C are lognormally distributed, and Ln(C) is normally distributed. One property of the normal distribution is that the sum of two normal distributions is a normal distribution, whose mean is the sum of the means and the variance the sum of the variances.

Mean (μ)

Variance (σ^2)

Ln(Q*C) = Ln(Q) * Ln(C)

$$\mu_{Q+C} = \mu_Q + \mu_C \qquad [5.9.a]$$

$$\sigma^2_{Q+C} = \sigma_Q^2 + \sigma_C^2 \qquad [5.9.b]$$

[5.9.c]

The probabilistic lognormal distributions describe the observed discharges well which are highly variable (2 orders of magnitude). Hence the natural logarithm of the pollutant load is the sum of two normally distributed variables Ln(Q) and Ln(C), and its distribution can be characterized by applying the relationships presented above. However, the parameters used to define concentrations and discharges are usually the median (50% of probability) and the coefficient of variation. Equations 5.9.b and 5.9.c use the mean and the variance. For a lognormal distribution, the mean and the variance can be obtained from the median and from the coefficient of variation by using the following relationships:

$$\text{Mean} = \text{Median} * \sqrt{(1+CV^2)} \qquad [5.10.a]$$

$$\text{CV} = \frac{\sigma}{\mu} \qquad [5.10.b]$$

However, extrapolating data to complete missing records prohibits the use of a probabilistic approach. Therefore average values are used, but their high temporal variability must be kept in mind. The objective is not to determine a value with a great accuracy but to give comparative, spatial intensive, information on a situation. The use of average discharges and loadings values is sufficient to meet this objective.

5.3.3 Calibration method

The discharge values measured at the gauging stations and computed by the model do not match perfectly; however they are of the same order of magnitude (Table 5.4). The objective of the calibration is to create, through a correction process, a perfect match at the USGS stations with the average observed flow.

The assumption underlying the correction process is that the error is uniformly distributed within each subwatershed: the errors are attributed to the runoff coefficients, both for base flow and for direct runoff. The recharge flow will not be corrected during the calibration process since it is based on observed values. Only the runoff coefficients must be multiplied by a correction factor so that the predicted discharges correspond to the observed ones.

- **Comparison with observed data**

The correction coefficient is first determined for the watersheds for which observed data are available (Figure 5.13).

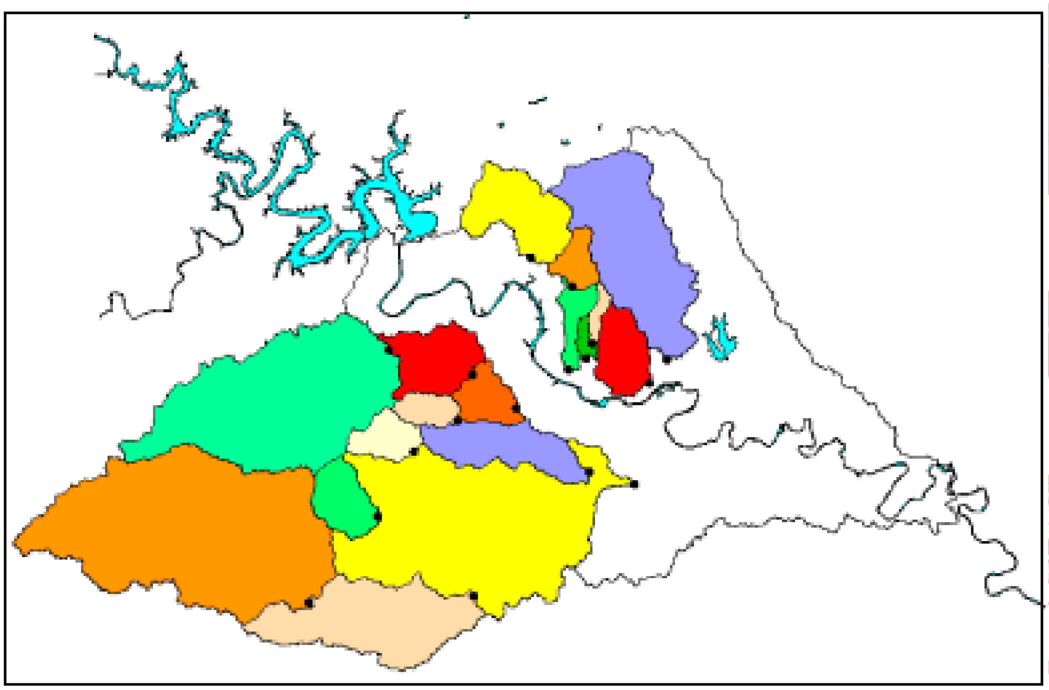


Figure 5.13: Calibration zones

An observed flow value (Q_{obs}) is supplied for each USGS watershed outlet, which is the location of the corresponding USGS station (Table 5.4). This value can be compared with the predicted discharge value at the same location (Q_{pred}). For nested watersheds, the comparison can be done for the flow generated within each watershed (Q_{zone_obs} and Q_{zone_pred}).

Table 5.4: Observed and predicted flow at the USGS stations (before flow correction)

Usgs#	Name	IC_{avg} (%)	Q_{obs} (cfs)	Q_{pred} (cfs)	Q_{zone_ob} (cfs)	Q_{zone_pred} (cfs)
8154700	Barton at Loop 360	16.6	60.3	59.9	-6.3	6.0
8156700	Barton at Lost Creek	9.0	66.6	53.9	12.6	9.3
8157000	Barton at SH 71	5.8	54.1	44.5	54.1	44.5
8155200	Bear at FM Road 1826	6.8	6.5	6.2	6.5	6.2
8157500	Boggy at US183	53.4	9.1	14.6	9.1	14.6
8158600	Bull at Loop 360	14.4	16.3	13.0	16.3	13.0
8156800	Onion at Buda	6.2	40.2	82.7	-22.0	22.0
8155240	Onion at Driftwood	5.1	62.2	60.7	62.2	60.7
8158050	Onion at US 183	11.0	91.9	170.5	26.9	59.1
8155300	Shoal at NW Park	58.3	4.7	7.8	4.7	7.8
8158920	Shoal at W 12th	54.3	7.7	14.4	3.0	6.7
8158840	Slaughter at FM Road 1826	12.9	5.8	5.0	5.8	5.0
8158970	Waller at 23rd	63.2	5.2	5.2	2.5	2.4
8159000	Waller at 38th	58.5	2.6	2.8	2.7	2.8
8158810	Walnut at Webberville Rd	28.9	35.6	41.6	35.6	41.6
8158800	Williamson at Oak Hill	16.1	4.7	3.7	4.7	3.7
8158700	Williamson at Jimmy Clay Rd	22.6	12.5	17.5	7.9	13.8

Negative Q_{zone_ob} indicate that more flow is lost to the recharge zone than generated by direct runoff and base flow in that zone (e.g. Barton at Loop 360 and Onion at Buda for which the recharge values are respectively 8.42 cfs and 31 cfs).

A correction coefficient is defined for each watershed by relating observed and predicted flow (without considering the recharge) according to the following relationship:

Observed flow = Predicted flow * flow correction – recharge [5.11]

In the case of nested watersheds, the incremental flow value must be considered as it represents the volume generated within the subpart of the watershed. The correction coefficient is defined as:

$$\text{flow correction} = \frac{\text{observed flow} + \text{recharge}}{\text{predicted flow (without recharge)}} \qquad [5.12]$$

The runoff coefficient/impervious cover relationships used can be evaluated by the study of the correction coefficients. A value of 1 would mean a perfect match, while values less than 1 (more than 1) shows that the runoff coefficients have been over-estimated (under-estimated). The values appear to be distributed from 0.348 (over-estimated) to 1.344 (under-estimated). However, neglecting certain values (shaded rows in Table 5.5) for stations within the recharge zone and for Waller whose hydrologic behavior can not be generalized, it appears that there is a link between the flow correction and the average impervious cover. In the case of nested watersheds, the average impervious cover is obtained by using the *zonalmean* function (section 4.4.2).

Table 5.5: Correction coefficients for gauged watersheds

Name	IC_{avg} (%)	Flow correction
Barton at Loop 360	16.6	0.348
Onion at Buda	6.2	0.409
Shoal at W 12th	54.3	0.448
Shoal at NW Park	58.3	0.603
Boggy at US183	53.4	0.626
Onion at US 183	11.0	0.667
Williamson at Jimmy Clay Road	22.6	0.707
Walnut at Webberville Road	28.9	0.855
Waller at 38th	58.5	0.933
Onion at Driftwood	5.1	1.025
Waller at 23rd	63.2	1.046
Bear at FM Road 1826	6.8	1.050
Slaughter at FM Road 1826	12.9	1.168
Barton at SH 71	5.8	1.214
Bull at Loop 360	14.4	1.249
Williamson at Oak Hill	16.1	1.254
Barton at Lost Creek	9.0	1.344

The points obtained by representing the correction coefficients versus the watershed-averaged impervious cover fit a linear function. The flow correction is higher than one for low impervious covers, which means that the runoff coefficients, and the discharge, have been underestimated. Similarly, the flow correction is less than one for high impervious covers, which means that the runoff coefficients and the discharge have been overestimated (Figure 5.14).

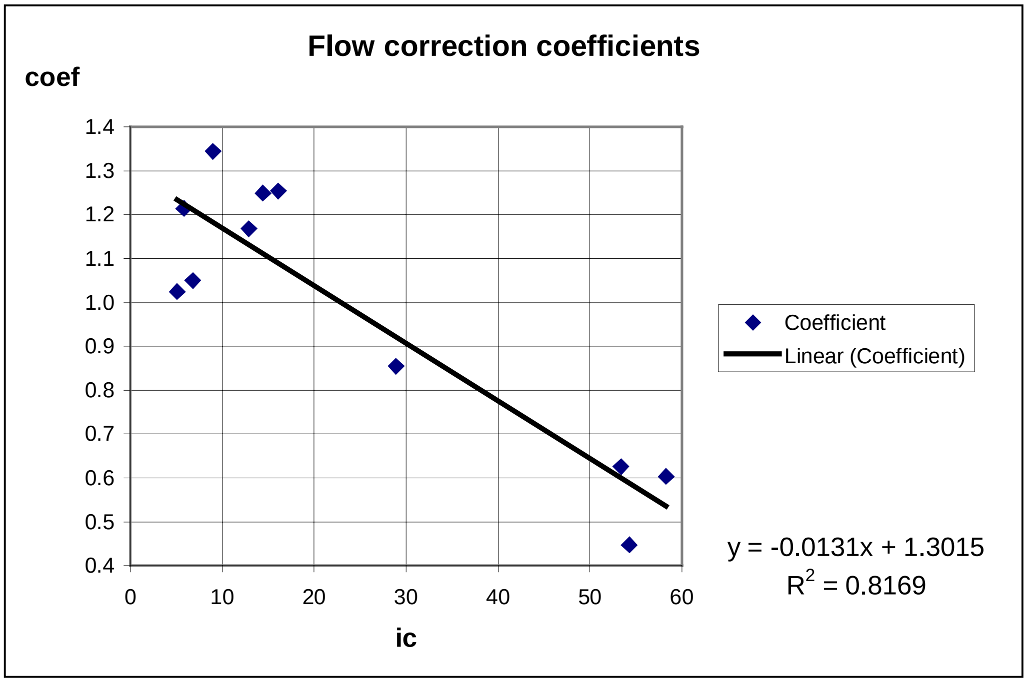


Figure 5.14: Relationship between flow correction and impervious cover

One explanation to this systematic error may be that the impervious cover has been defined in two different ways:

- The first definition (Environmental and Conservation Services Department) was used to establish the relationships between impervious cover and other parameters (runoff, EMC) using data from water quality sampling sites.
- A second definition (Planning and Development Department) was used to relate land use and impervious cover with maps for the whole city.

It would be useful in the future if a more consistent measure of impervious cover were use by the two Departments.

- **Flow corrections for watersheds without data**

The flow correction coefficients are calculated for watersheds with sufficient USGS data. Extrapolation is necessary for watersheds without data. The data has shown that, except for watersheds which have unique characteristics (e.g. recharge zone, Waller Creek), a linear relationship can be established between the correction coefficient and impervious cover (Figure 5.14).

Extrapolated flow correction coefficient = - 0.0131 * IC(%) + 1.3015 [5.13]

For high impervious cover, equation 5.13 gives negative values. Because this relationship was established with average impervious cover over large watersheds, the average impervious cover is used in this equation. This ensures that negative values do not occur.

- **Generate a correction grid**

A flow correction grid is created which contains:

1. the correction coefficients directly defined for the subwatersheds for which observed values are available (the downstream end of a watershed in which USGS subwatersheds are located takes the correction coefficient of the USGS subwatershed immediately next to him (Figure 5.15).
2. otherwise use the extrapolated correction coefficients for ungaged areas.

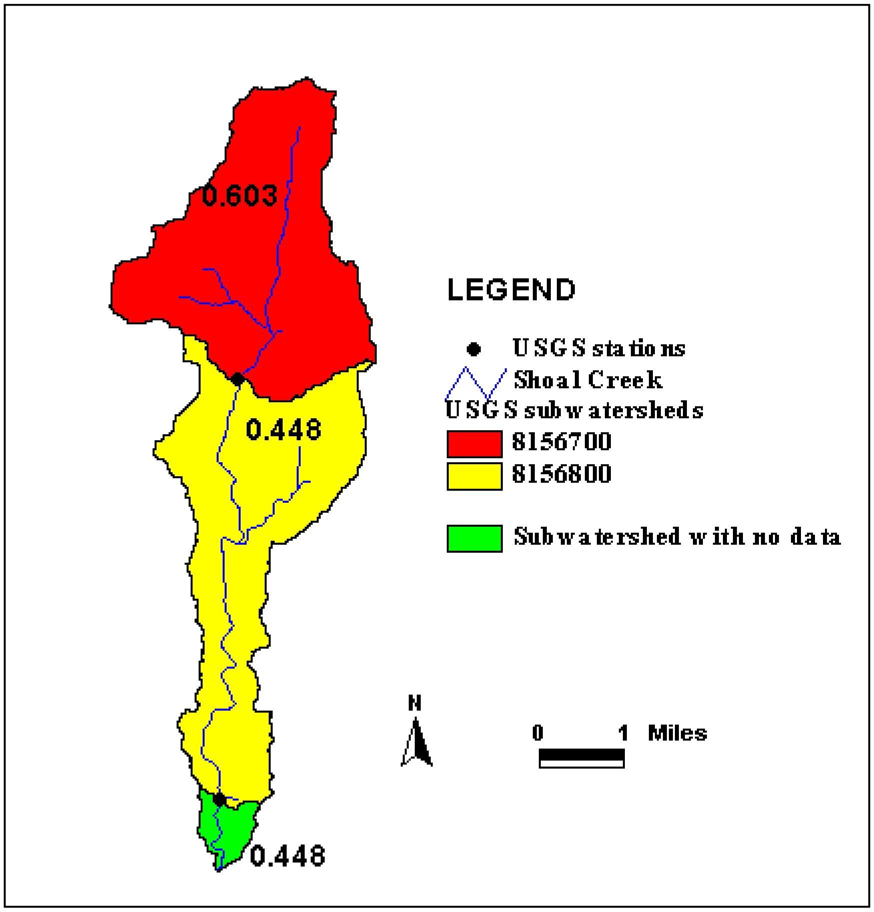


Figure 5.15: Correction coefficients

Step 1: Apply the extrapolated flow corrections relationship

First an extrapolated flow correction grid is created by applying equation 5.13 to a grid whose cell value is the correction corresponding to the average impervious cover of the watershed in which the cell is located. The average impervious cover grid (*icavgwshd*) is created with the function *zonalmean* as shown below in Arc/Info, or by using the script *Qual.Zonalmean* in ArcView (section 4.2.2). A correction coefficient is hence defined for each watershed.

Grid: **icavgwshd = zonalmean (coawshd_gr , ic_gr)**
Grid: **corcoef1 = -0.0131 * icavgwshd + 1.3015**

Step 2: Downstream end of watersheds with USGS data

The subwatersheds for which no data area available but which are located downstream of a USGS watershed are assigned the correction calculated for that USGS watershed. If a correction coefficient has been calculated for a portion of a watershed (i.e. for Barton, Onion, Waller and Bull), the whole watershed is assigned the correction coefficient corresponding to the most downstream subwatershed. The procedure which modifies the correction in Onion is described below. It must be done for each watershed.

For Onion:
Grid-code = 36 in *coawshd_gr*
Correction coefficient: 0.667

Grid: **corcoef21 = con (coawshd_gr == 36 , 0.667 , corcoef1)**

The different flow correction coefficients to use for each watershed are shown in Table 5.6.

Table 5.6: Correction coefficients for downstream portions of USGS watersheds

Watersheds	Coawshd_gr grid code	Coefficient
Barton	19	0.348
Boggy	20	0.626
Bull	3	1.249
Onion	36	0.667
Shoal	6	0.448
Waller	11	1.046
Walnut	2	0.855

Step 3

The last step is to assign the directly calculated correction values to the USGS watersheds. The Grid function *reclass* allows one to use the one to one relationship between the USGS number and the correction coefficients to create the correction grid (*zone_corr*) from the USGS watersheds grid: this grid contains only the USGS watersheds. Each cell is assigned the correction coefficient associated to the USGS number (Table 5.7). The function *reclass* can only handle integers: the coefficients have first to be multiplied by 1,000 to meet this criterion.

Grid: **zone_corr = reclass (zones , corrcoef.txt)**

where corrcoef.txt is the file associating USGS number and correction coefficients * 1,000.

Table 5.7: USGS number/Correction coefficient relationship

8154700 : 1249
8155200 : 1214
8155240 : 1344
8155300 : 348
8156700 : 603
8156800 : 448
8157000 : 933
8157500 : 1046
8158050 : 626
8158600 : 855
8158700 : 1025
8158800 : 409
8158810 : 1050
8158840 : 1168
8158920 : 1254
8158970 : 707
8159000 : 667

The resulting grid is multiplied by 10^{-3} to obtain the real coefficients. As the input grid is an integer grid, dividing by 1,000 would have created an integer grid with 0 or 1 as cell value.

Grid: **zone_corr1 = 0.001 * zone_corr**

Finally this grid is merged on top of the grid obtained at the end of the step 2 (*corcoef27*).

Grid: **corcoef = merge (zone_corr1 , corcoef27)**

This correction coefficient grid is used in all the future discharge and load computations. The calibrated runoff coefficients for direct runoff and for base flow are obtained by multiplying a grid with the original runoff coefficients by the correction coefficient grid *corcoef*.

5.4 DISCHARGE COMPUTATION

Recharge and flow corrections were determined from observed data. They are used in the extrapolation process to future conditions assuming that:

- the recharge zone and recharge rate do not vary.
- the land use/impervious cover relationships remain the same.
- the runoff coefficient/impervious cover relationships remain the same.
- the precipitation value used is 31.08 inches/year.

Any variations in these assumptions require modification of the calibration process. The calibration parameters defined here (recharge and correction coefficients) are valid only under these assumptions. It is however certain that the relationships will vary in the future, due in part to the availability of additional data. This means that the calibration data is necessary at that point.

Once the calibration has been done, the computation of the discharge from the impervious cover is straightforward. Either an AML (Procedure 5.2, Flow.aml) or an Avenue script (*Qual.Flow*) can be used to do the computation. They are based on the same principle and differ only by the language used. An AML allows executing the commands in series automatically, instead of having to type each line in Arc/Info. The AML also allows one to summarize the process presented here. The language used in the Avenue script is slightly different. A description of the Avenue script is given in Chapter 8 and Appendix C.

Procedure 5.2: Discharge computation (with flow corrections)

```
/*--------------FLOW.AML --------
/*-----------------------------------------------
/*FUNCTION: compute the discharge
/*INPUT DATA: impervious cover grid (icfut_gr), flowdirection grid (burn_fdr),
correction grid (corcoef), precipitation (31.08 in/yr), recharge grid (rech_fac), recharge
generated in each cell (lcorr_rech)
/*-----------------------------------------------
/*BEGIN AML EXECUTION

grid
setcell 100

/*Compute the runoff coefficients for direct runoff
runcoef = 0.3428 *icfut_gr * icfut_gr + 0.5677 * icfut_gr + 0.0125

/*Compute the runoff coefficients for base flow (must be positive)
bflowcoef0 = -0.36 * icfut_gr + 0.1904
bflowcoef = con ( bflowcoef0 < 0 , 0 , bflowcoef0 )
kill bflowcoef0 all


/*Compute the corrected direct runoff generated by each cell (cfs)
runcell = runcoef * 31.08 * 5 * corcoef / 189216
kill runcoef all

/*Compute the corrected base flow generated by each cell (cfs)
bflowcell = bflowcoef * 31.08 * 5 * corcoef / 189216
kill bflowcoef all

/*Compute the total direct runoff, base flow and total flow in each cell in cfs (without
considering recharge)
runoff0 = flowaccumulation (burn_fdr , runcell )
baseflow0 = flowaccumulation ( burn_fdr , bflowcell )
totalflow0 = runoff0 + baseflow0

/*Compute the total flow and base flow with the recharge
flow_pred = totalflow0 - rech_fac
partbaseflow = baseflow0 / totalflow0
rechbfcell = lcorr_rech * partbaseflow
rechbf_fac = flowaccumulation ( burn_fdr , rechbfcell )
baseflow = baseflow0 - rechbf_fac
runoff = flow_pred - baseflow
&return
/*--------------- END OF AML --------------------
```

Chapter 6: Load

Non-point source pollution can be divided into a land (external) and an in-stream load component. The land load corresponds to the load generated by the water moving on or through the ground while the in-stream load is the result of channel erosion. The external load is computed directly by using expected mean concentrations in runoff. The channel erosion rate is inferred by computing the difference between observed loads and those predicted using land surface loads alone.

6.1 External load

6.1.1 Event mean concentration

- **Definition**

Constituent concentrations associated with each cell are related to the land use. For each cell, there are two concentration values, one for direct runoff and the other for base flow. Mean concentrations are computed from storm event data (event mean concentrations). The event mean concentration for a storm is the total storm load (mass) divided by the total runoff volume.

$$EMC = \overline{C} = \frac{M}{V} = \frac{\int \mathrm{C(t)Q(t)\ dt}}{\int \mathrm{Q(t)\ dt}} \qquad [6.1]$$

The loading is estimated by multiplying the EMC by the runoff volume. The actual instantaneous concentrations observed during a storm may be higher or lower than the EMC. As EMCs vary from storm to storm, a median or 50^{th} percentile is defined to characterize a site's EMC. The variability between different sites and events is quantified by a median and by a coefficient of variation. Table 6.1 shows the relationships used in this study: they are defined for an impervious cover defined as a decimal fraction (0<IC<1).

Table 6.1: Impervious cover/ EMC relationships

Constituent	Direct runoff concentration (mg/l)*	Base flow concentration in undeveloped areas (mg/l)**	Base flow concentration in developed areas (mg/l)
TSS	190	0	0
BOD	C=14(IC)+3.5	0.45	0.8
COD	C=98(IC)+18	12	20
TOC	C=8.6(IC)+8	2	5
DP	C=0.24(IC)+0.04	0.014	0.06
TP	C=0.32(IC)+0.19	0.02	0.12
NH_3	C=0.24(IC)+0.13	0.02	0.06
TKN	C=1.53(IC)+0.13	0.28	0.46
NO_3	0.82	0.15	0.6
TN	1.53(IC)+0.95	0.43	1.06
Cu	C=0.016(IC)+0.006	NA	NA
Pb	C=0.038(IC)+0.003	NA	NA
Zn	C=0.19(IC)	NA	NA

* Zero EMCs are associated with water land use.
**Areas with an impervious cover less than 15% are considered undeveloped.
IC Impervious Cover expressed as a decimal fraction (0<IC<1).

These relationships are defined in the report "Water Quality and Quantity Inputs for the Urban Creeks Future Needs Assessment", by Michael E. Barrett (July 1997). The values for Total Nitrogen correspond to the sum of the EMCs for TKN and NO_3.

- **EMC grids**

Using the relationships in Table 6.1, the EMC grids for direct runoff and base flow can be created by either of two ways.

(a) Starting with a land use or traffic serial zone coverage

The EMC field for each constituent is computed in the land use coverage attribute table by creating an EMC field for each constituent and applying the IC/EMC relationships from Table 6.1. It is necessary to use parentheses when typing the equations since ArcView does the computation from left to right. There is no priority given to multiplication over addition. For example, the expression a+b*c is computed in ArcView as (a+b)*c instead of a+(b*c), as normally dictated by mathematical rules.

The 100% impervious cover associated with water yields high EMCs values for the water land use. As rainfall contributes directly to the water body, zero EMCs should be associated with water. The EMCs obtained by applying the relationships shown in Table 6.1 must be corrected for the water land use. The records whose EMCs must be set to zero must be selected in the attribute table. For current conditions, water corresponds to the land use code 940 (*newlandusecode* = 940). For future conditions, water zones are represented by the traffic serial zones 999 (*tsz* = 999). The function *polygrid* is used to create the EMC grids for direct runoff and base flow by using the corresponding fields in the attribute table (e.g. *bod_emc* for BOD direct runoff EMCs and *bod_emcbf* for BOD base flow EMCs).

Grid: **setcell 100** *(set the cell size to 100ft)*
Grid: **emc_gr = polygrid (finluse , bod_emc)**
Grid: **emcbf_gr = polygrid (finluse , bod_emcbf)**

(b) Starting with an impervious cover grid

The other possibility is to start with an impervious cover grid created with the command *polygrid* from the land use coverage. The IC/EMC relationships in Table 6.1 are applied to that grid in order to create EMC grids. The EMCs in the water zones must be set to zero. For future conditions, these zones are characterized by the grid-code 999 (traffic serial zone number) in the traffic serial zones grid *zone_gr*.

Build the traffic serial zones grid
Grid: **zone_gr = polygrid (zones , tsz)**

Apply the IC/EMC relationship to the impervious cover grid icfut_gr (e.g. BOD)
Direct runoff
Grid: **emc_gr0 = 14 * icfut_gr + 3.5**
Base flow
Grid: **emcbf_gr0 = con (icfut_gr <= 0.15 , 0.45 , 0.8)**

Correct the water zones
Grid: **emc_gr = con (zone_gr == 999 , 0 , emc_gr0)**
Grid: **kill emc_gr0 all**
Grid: **emcbf_gr = con (zone_gr == 999 , 0 , emcbf_gr0)**
Grid: **kill emcbf_gr0 all**

6.1.2 Load computation method

- **Load produced by each cell**

The calibrated discharge and concentration grids for direct runoff and base flow are multiplied together to generate the direct runoff (Figure 6.1) and base flow loads produced in each cell. The total load produced in each cell is obtained by adding the contributions of base flow and direct runoff.

- Load produced by direct runoff

$Load_r$ (kg/year) = EMC(mg/l) * Q_r (cfs) * $(3.048)^3$(l/ft^3) * 86400(s/d) * 365(d/yr) * 10^{-6}(kg/mg)

Grid**: loadcell = emc_gr * runcell * 3.048 * 3.048 * 3.048 * 86400 * 365 / 1000000**

- Load produced by baseflow

$Load_{bf}$(kg/year) = EMC_{bf}(mg/l) * Q_{bf}(cfs) * $(3.048)^3$(l/ft^3) * 86400(s/d) * 365(d/yr) * 10^{-6} (kg/mg)

Grid: **loadcellbf = emcbf_gr * bflowcell * 3.048 * 3.048 * 3.048 * 86400 * 365 / 1000000**

- Total load

$Load_t$ (kg/year) = $Load_r$ + $Load_{bf}$

Grid: **tloadcell = loadcell + loadcellbf**

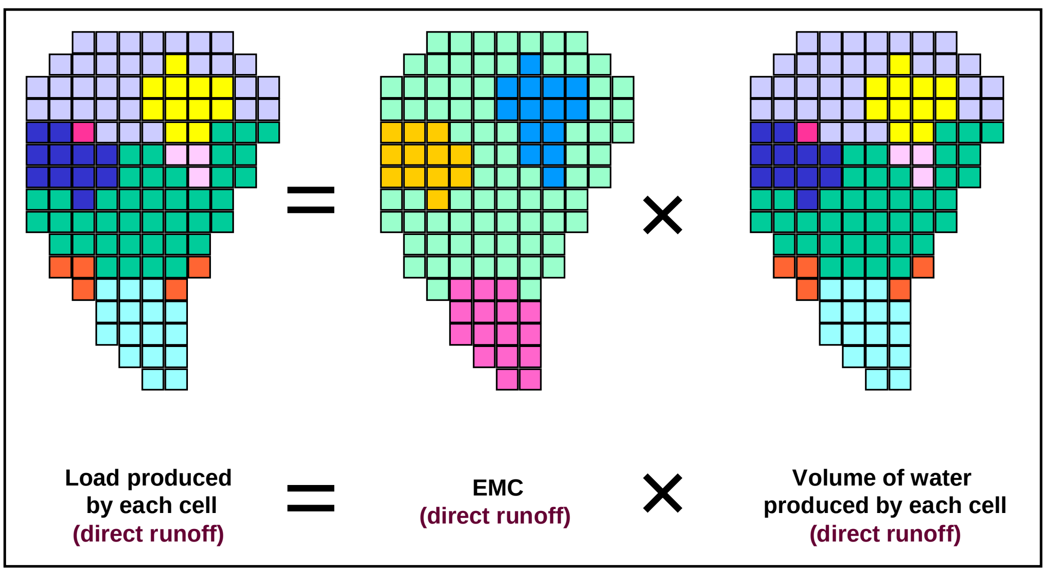


Figure 6.1: Load produced by each cell (e.g. direct runoff)

- **Total load (without recharge)**

The total load in any cell is obtained by summing the load contribution of all cells located within the drainage area upstream of the given cell. This procedure is similar to the discharge computation which uses a weighted flowaccumulation in Arc/Info Grid.

Grid**: load0 = flowaccumulation (burn_fdr , tloadcell)**

- **Recharge load**

Like the difference between the flow generated and lost in the cells located in the creeks within the recharge zone, the difference between the load generated and lost can be negative. The problem lies in the fact that the *flowaccumulation* function cannot handle negative values. If the grid is entirely negative, the solution is to take its absolute value and then run the *flowaccumulation* function and take its negative value. However, for this research both positive and negative load values are found in the grid. The load lost in the recharge zone must be computed separately and subtracted from the total load.

Concentrations are obtained by dividing load by flow. The concentrations in the water lost in a cell in the recharge zone are assumed to be the same as the concentrations in the creek at the cell where the recharge occurs. They are the same as the concentrations in the creek without considering the recharge zone (*co*), which are obtained by dividing the total load (*load0*) by the total flow (totalflow0). The load lost in each cell of the recharge zone is the product of this concentration and the volume of flow lost in the cells within the recharge zone (*lcorr_rech*). The *flowaccumulation* function is then used to add the contribution of all the cells to recharge load losses.

Grid: **co = load0 / totalflow0**
Grid: **loadcellrech = lcorr_rech * co**
Grid: **loadrech = flowaccumulation (burn_fdr , loadcellrech)**

The load lost to the recharge zone is finally subtracted from the total load previously computed (*load0*) to obtain the external load (*newload*).

Grid: **load = load0 - loadrech**

- **Load computation**

The different steps presented above can be gathered in a program, either in Arc/Info or ArcView (chapter8). The AML shown below (procedure 6.1) allows the user to compute the external load directly from an impervious cover grid once the discharge has been computed. This AML is an example of the load computation for BOD. It must be edited for each constituent.

Procedure 6.1: Load computation

```
/* ----------------------------------------------
/*-----------------------------------------------
/*-------------- LOAD.AML ----------------
*------------------------------------------------
/*-----------------------------------------------
/*FUNCTION: compute the load given an impervious cover and a runoff coefficient
coverages.
/*INPUT GRIDS: impervious cover (icfut_gr, 0<IC<1), traffic serial zones (zone_gr),
direct runoff (runcell) and base flow (bflowcell) generated in each cell (in cfs),
flowdirection (burn_fdr), total flow without recharge (totalflow0), recharge lost in each
cell (lcorr_rech).

/*-----------------------------------------------
/*BEGIN AML EXECUTION

grid
setcell 100

/*Compute the EMC for direct runoff and base flow
emc_gr0 = 14 * icfut_gr + 3.5
```

```
emc_gr = con ( zone_gr == 999 , 0 , emc_gr0 )
kill emc_gr0 all
emcbf_gr0 =  con ( icfut_gr <= 0.15 , 0.45 , 0.8 )
emcbf_gr = con ( zone_gr == 999 , 0 , emcbf_gr0 )
kill emcbf_gr0 all

/*Compute the load produced by each cell
loadcell = runcell * emc_gr * 3.048 * 3.048 * 3.048 * 86400 * 365 / 1000000
loadcellbf = bflowcell * emcbf_gr * 3.048 * 3.048 * 3.048 * 86400 * 365 / 1000000
tloadcell = loadcell + loadcellbf
kill loadcell all
kill loadcellbf all


/*Compute the total load (without considering the recharge zone)
load0 = flowaccumulation ( burn_fdr , tloadcell )
kill tloadcell all

/*Compute the recharge load
BOD_co = load0 / totalflow0
loadcellrech = lcorr_rech * BOD_co
loadrech = flowaccumulation ( burn_fdr , loadcellrech )
kill loadcellrech all


/*Compute the total load
BOD_load = load0 - loadrech
kill load0 all
kill loadrech all

&return

/*--------------- END OF AML --------------------
```

- **Measured versus predicted values**

Concentration

The predicted land surface loads are divided by the predicted discharges, converted to concentrations in mg/l, and compared with the concentrations measured at the USGS stations (Figures 6.2.a-6.2.j). No measured concentrations are available for copper (Cu), lead (Pb) and zinc (Zn).

$$\text{Predicted concentration [mg/l]} = \frac{\text{Predicted Load [kg/yr]} * 10^6 [mg / kg]}{\text{Predicted Flow [cfs]} * (3.048[l / ft^3]) * 86400[s / d] * 365[d / yr]}$$

[6.2]

The graphs show that the predicted values for BOD (6.2.a), COD (6.2.c), DP (6.2.e), NH_3 (6.2.i) and NO3 (6.2.j) match well the measured ones. The concentrations for TOC (6.2.b), TP (6.2.d), TSS (6.2.f), TKN (6.2.g) and TN (6.2.h) appear however constantly underestimated.

Load

The study deals more with loads than with concentration. "Measured" loads can be estimated by multiplying the measured concentrations by the predicted flows. Even though they are not really measured loads, these loads are used to evaluate the predicted loads. Since the discharge value varies for each station, the graphs comparing the loads (Figures 6.3.a-6.3.j) do not look exactly the same as the graphs comparing the concentrations. The general trend is the same however: TOC, TP, TSS, TKN and TN are underestimated.

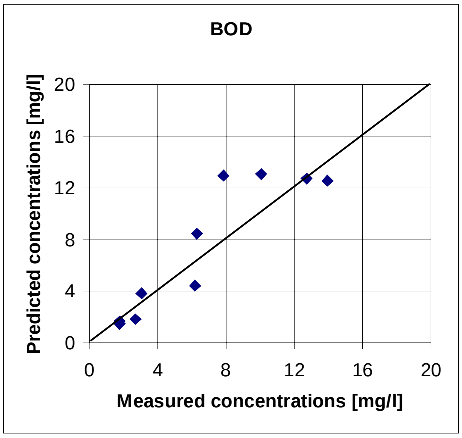


Figure 6.2.a

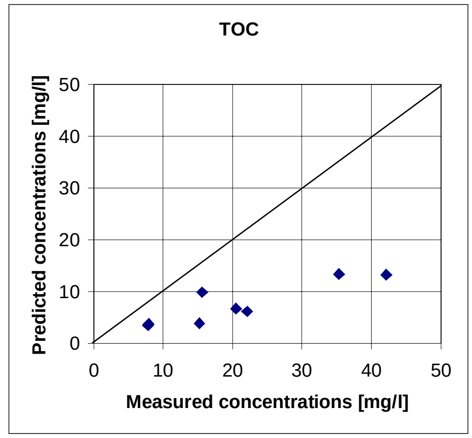


Figure 6.2.b

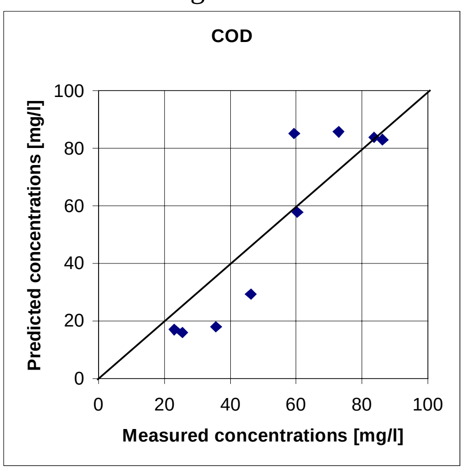


Figure 6.2.c

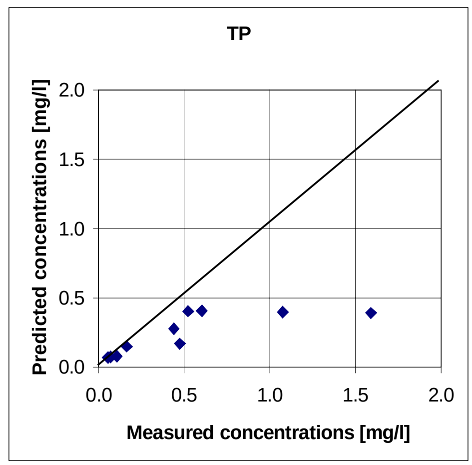


Figure 6.2.d

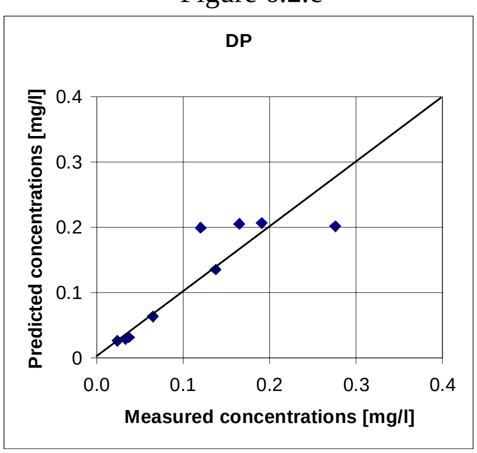


Figure 6.2.e

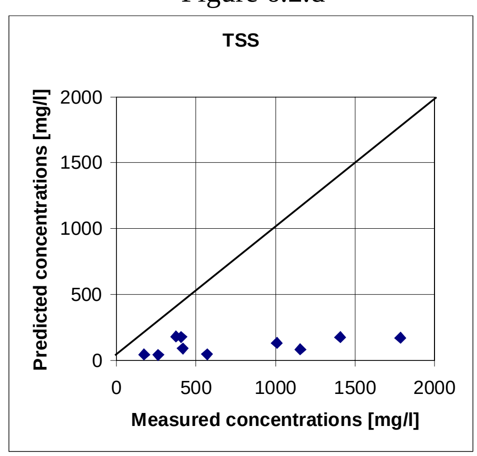


Figure 6.2.f

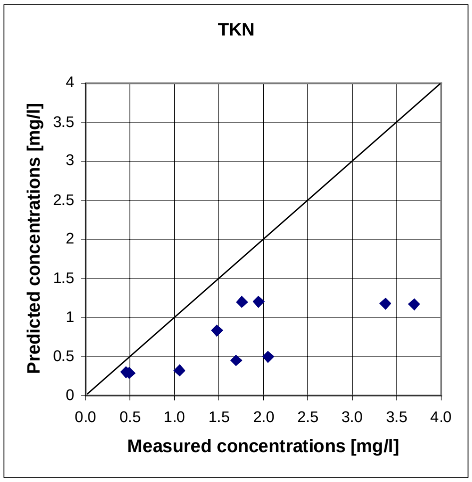


Figure 6.2.g

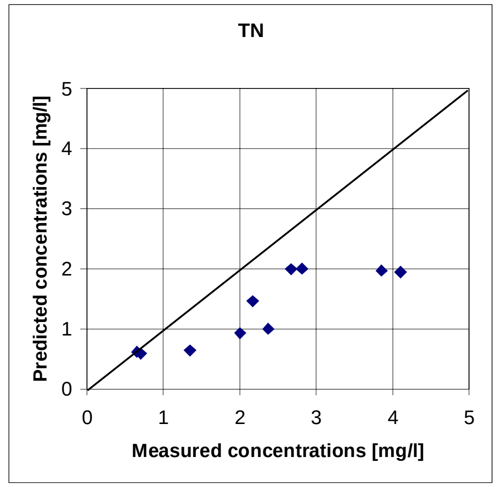


Figure 6.2.h

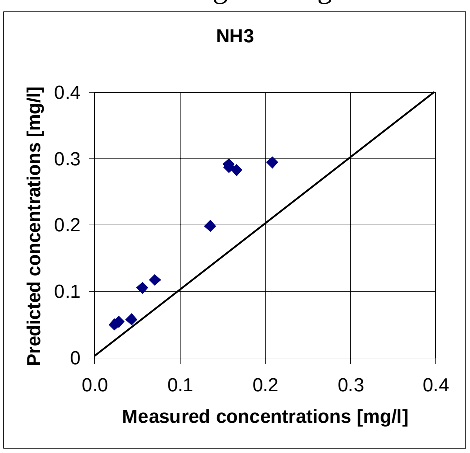


Figure 6.2.i

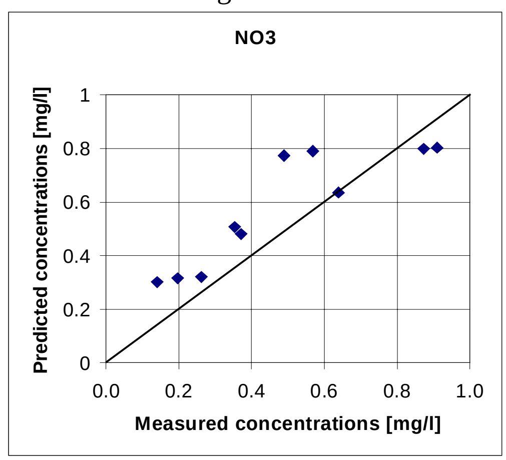


Figure 6.2.j

Figures 6.2: Comparison between measured and predicted concentrations at the USGS stations

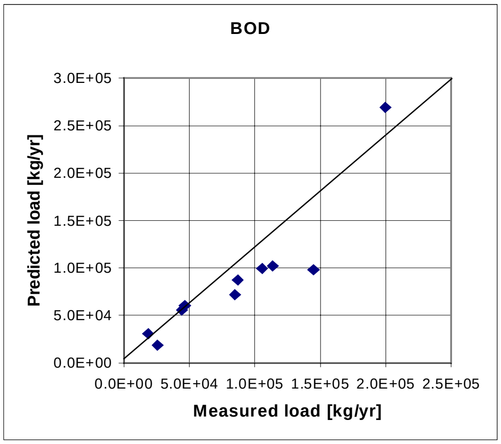


Figure 6.3.a

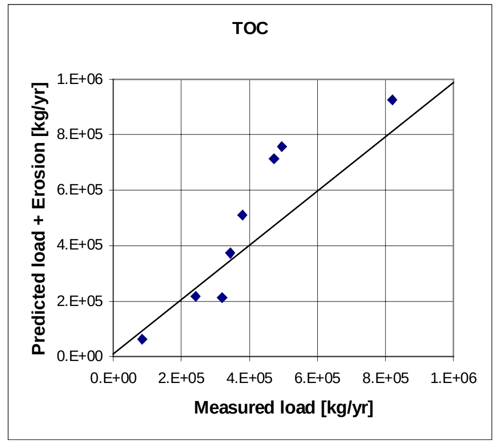


Figure 6.3.b

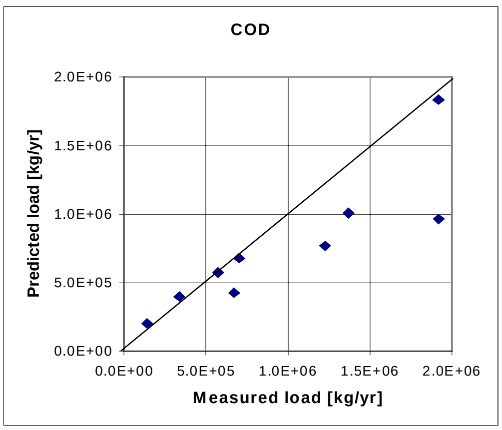


Figure 6.3.c

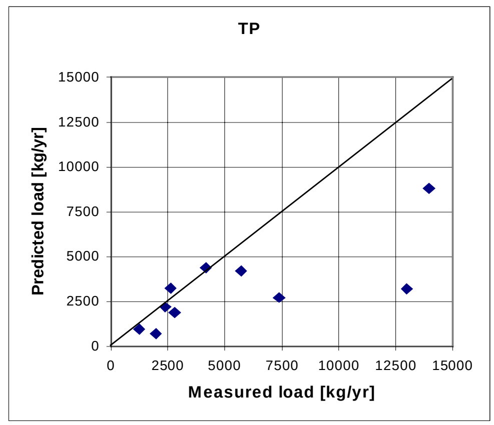


Figure 6.3.d

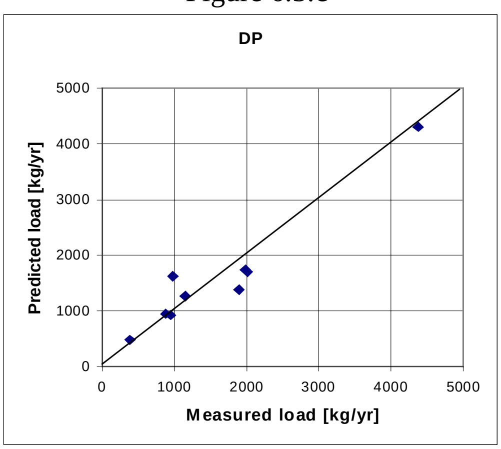


Figure 6.3.e

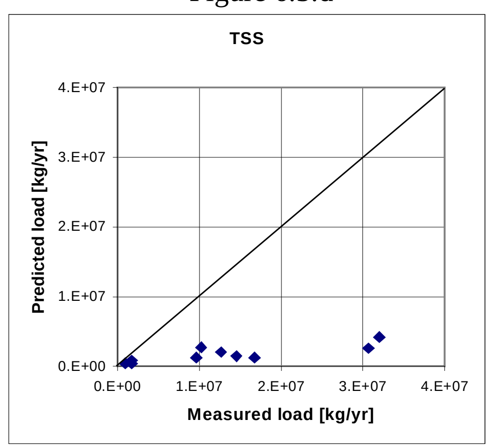


Figure 6.3.f

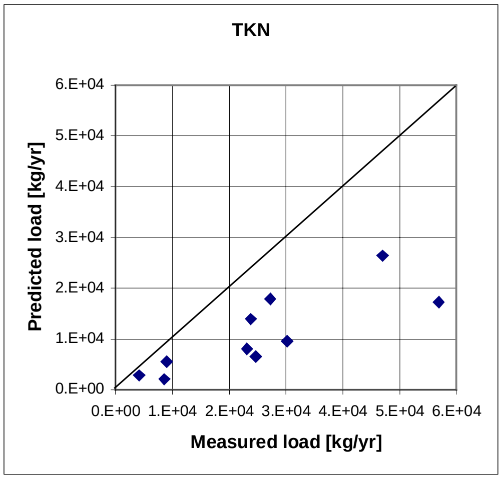


Figure 6.3.g

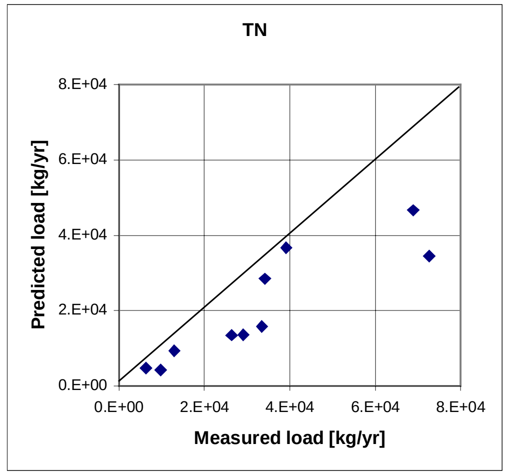


Figure 6.3.h

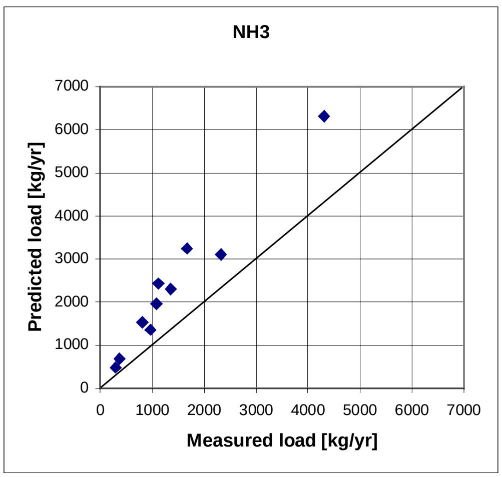


Figure 6.3.i

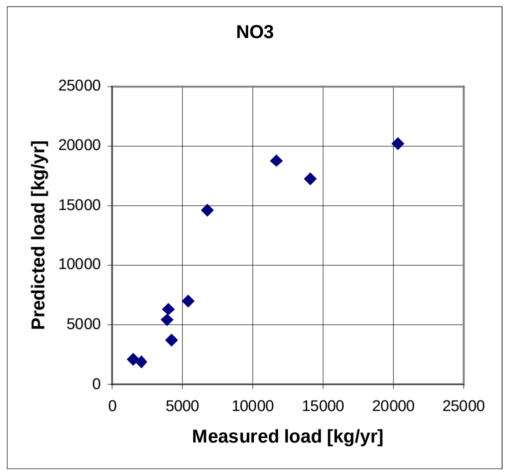


Figure 6.3.j

Figures 6.3: Comparison between measured and predicted loads at the USGS stations

Predicted loads are underestimated because they do not take into account the in-stream loads due to erosion which can be very important (e.g. TSS).

6.2 CHANNEL EROSION

6.2.1 Model

The second component of non-point source pollution is the in-stream channel erosion, which is difficult to evaluate directly: it is computed as the difference between the "measured" and the predicted loads. The effect of the current Best Management Practices on the loads is assumed to be negligible compared with the uncertainties in other data (e.g. impervious cover). The effects of current BMPs are not taken into account in the erosion computation.

Erosion happens in and aroundthe channel. An erosion coefficient, representing the load eroded by length of channel can be defined for each watershed where measured data are available. This coefficient is defined as:

$$\text{Erosion coefficient [kg/yr - ft]} = \frac{(\text{observed load - predicted load})_{\text{at the station}}}{\text{channel length upstream of the station}} \qquad [6.3]$$

- **Channel definition**

It is necessary to define the channel where in-stream erosion occurs. There are two alternatives. The first one is to consider that the channels correspond to the digitized creeks. The second consists of choosing a flowaccumulation threshold: any cell whose drainage area is bigger than the threshold value is located in a channel. The problem however is to define the threshold.

As the digitized creeks are the result of field observation, they should give a good representation of the channels. It is not possible to find a flowaccumulation threshold value for which all delineated creeks correspond to the digitized creeks. The channel length is hence based on the digitized creeks. The channel length is computed by using the flowdirection grid: a length value related to the direction of the flow is associated with each cell (section 5.2). The delineated creeks, which are based on the flowdirection grid, do not correspond exactly to the digitized creeks. Note that the channel length at any location is hence slightly different from the length which would have been defined by using the delineated creeks derived from the digital elevation model.

6.2.2 Erosion coefficients

An erosion coefficient is defined for each USGS watershed for which data are available. Erosion is caused by the in-stream flow, which is based on the average impervious cover. Linear relationships between erosion coefficient and average impervious cover were hence defined to extrapolate the erosion to the watersheds with no observed data. Erosion is considered only for the constituents whose predicted load was underestimated:

- Total Organic Carbon (Table 6.2, Figure 6.4)
- Total Nitrogen (Table 6.3, Figure 6.5, erosion for TKN is not considered)
- Total Phosphorus (Table 6.4, Figure 6.6)
- Total Suspended Solids (Table 6.5, Figure 6.7)

Total Organic Carbon

Table 6.2: TOC erosion coefficients

USGS Station	Channel length ft	Discharge cfs	TOC obs kg/yr	TOC pred kg/yr	Erosion kg/yr.ft	IC %
Barton at Loop 360	641015	60.3	818933	206397	0.96	16.6
Barton at Lost Creek	577037	66.6	472801	220249	0.44	9
Barton at SH 71	455497	54.1	378424	168903	0.46	5.8
Boggy at US183	110232	9.1	343657	107744	2.14	53.4
Bull at Loop 360	117563	16.3	320911	89223	1.97	14.4
Shoal at W 12th	52248	7.7	241999	91528	2.88	54.3
Walnut at Webberville rd	287253	35.6	496217	313209	0.64	28.9
Williamson at Oak Hill	32454	4.7	85207	27647	1.77	16.1

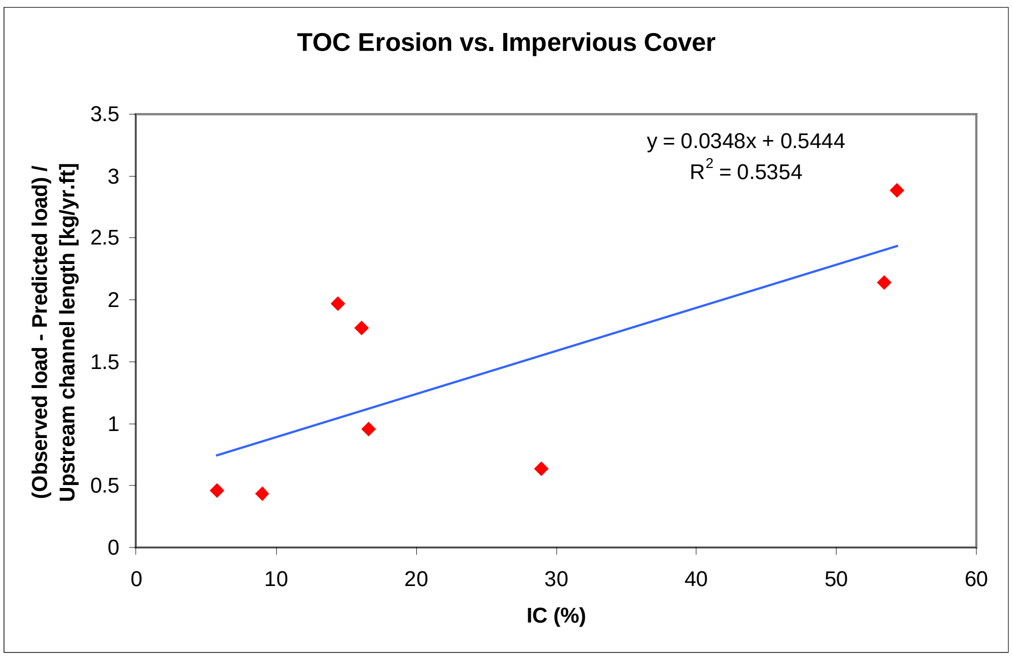


Figure 6.4: TOC erosion vs. impervious cover

Total Nitrogen

Table 6.3: TN erosion coefficients

USGS Station	Channel length ft	Discharge cfs	TN obs kg/yr	TN pred kg/yr	Erosion kg/yr.ft	IC %
Barton at Loop 360	641015	60.3	72649	34549	5.94E-02	16.6
Barton at Lost Creek	577037	66.6	39147	36677	4.28E-03	9
Barton at SH 71	455497	54.1	34180	28487	1.25E-02	5.8
Boggy at US183	110232	9.1	33465	15860	1.60E-01	53.4
Bull at Loop 360	117563	16.3	29129	13522	1.33E-01	14.4
Shoal at W 12th	52248	7.7	26404	13479	2.47E-01	54.3
Walnut at Webberville rd	287253	35.6	68894	46604	7.76E-02	28.9
Williamson at Oak Hill	32454	4.7	9851	4169	1.75E-01	16.1

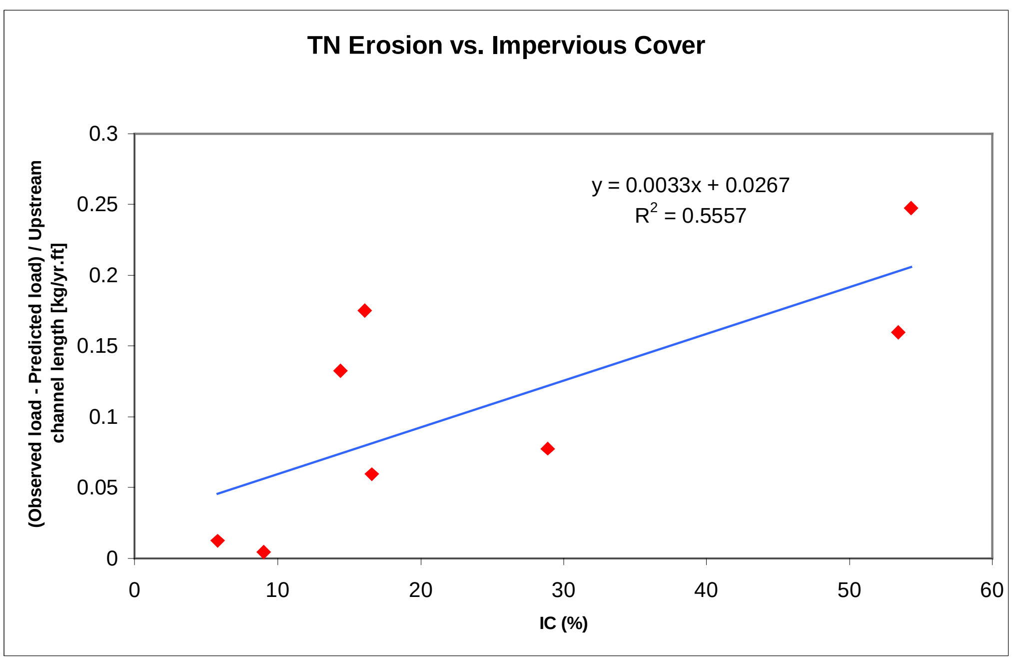


Figure 6.5: TN erosion vs. impervious cover

Total Phosphorus

Table 6.4: TP erosion coefficients

USGS Station	Channel length ft	Discharge cfs	TP obs kg/yr	TP pred kg/yr	Erosion kg/yr.ft	IC %
Barton at Loop 360	641015	60.3	5710	4205	2.35E-03	16.6
Barton at Lost Creek	577037	66.6	4166	4393	-3.94E-04	9
Barton at SH 71	455497	54.1	2615	3257	-1.41E-03	5.8
Boggy at US183	110232	9.1	12982	3186	8.89E-02	53.4
Bull at Loop 360	117563	16.3	2375	2179	1.66E-03	14.4
Shoal at W 12th	52248	7.7	7368	2719	8.90E-02	54.3
Walnut at Webberville rd	287253	35.6	13963	8814	1.79E-02	28.9
Williamson at Oak Hill	32454	4.7	1966	695	3.92E-02	16.1

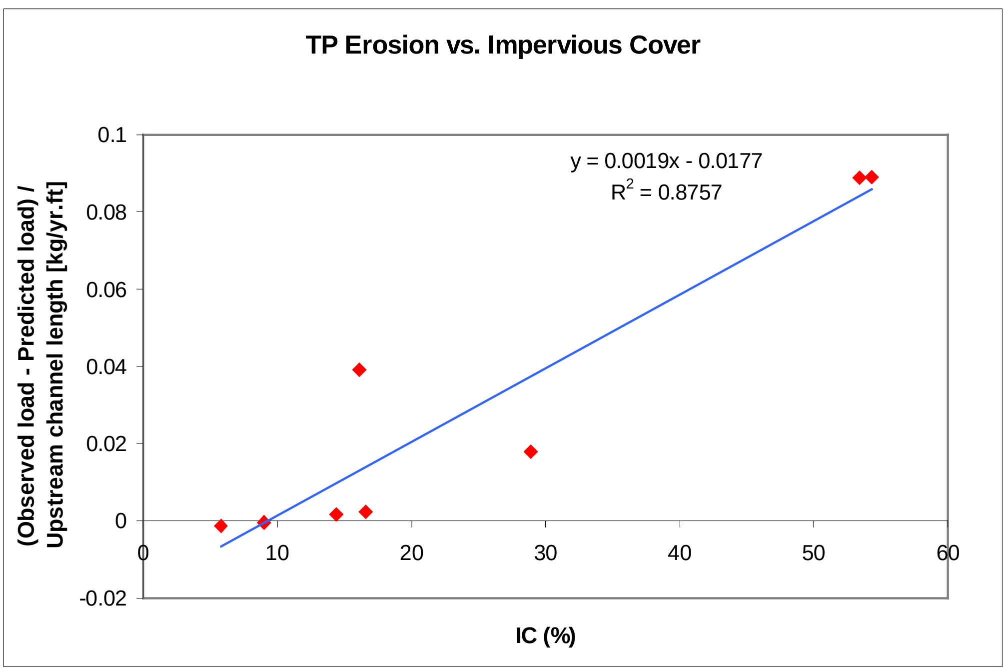


Figure 6.6: TP erosion vs. impervious cover

Note that for low impervious covers the relationship yields a negative erosion (e.g. Barton).

Total Suspended Solids

Table 6.5: TSS erosion coefficients

USGS Station	Channel length ft	Discharge cfs	TSS obs kg/yr	TSS pred kg/yr	Erosion kg/yr.ft	IC %
Barton at Loop 360	641015	60.3	3.07E+07	2.53E+06	44	16.6
Barton at Lost Creek	577037	66.6	1.03E+07	2.69E+06	13	9
Barton at SH 71	455497	54.1	1.27E+07	2.03E+06	23	5.8
Boggy at US183	110232	9.1	1.45E+07	1.40E+06	119	53.4
Bull at Loop 360	117563	16.3	1.68E+07	1.22E+06	132	14.4
Shoal at W 12th	52248	7.7	9.63E+06	1.21E+06	161	54.3
Walnut at Webberville rd	287253	35.6	3.20E+07	4.16E+06	97	28.9
Williamson at Oak Hill	32454	4.7	1.73E+06	3.77E+05	42	16.1

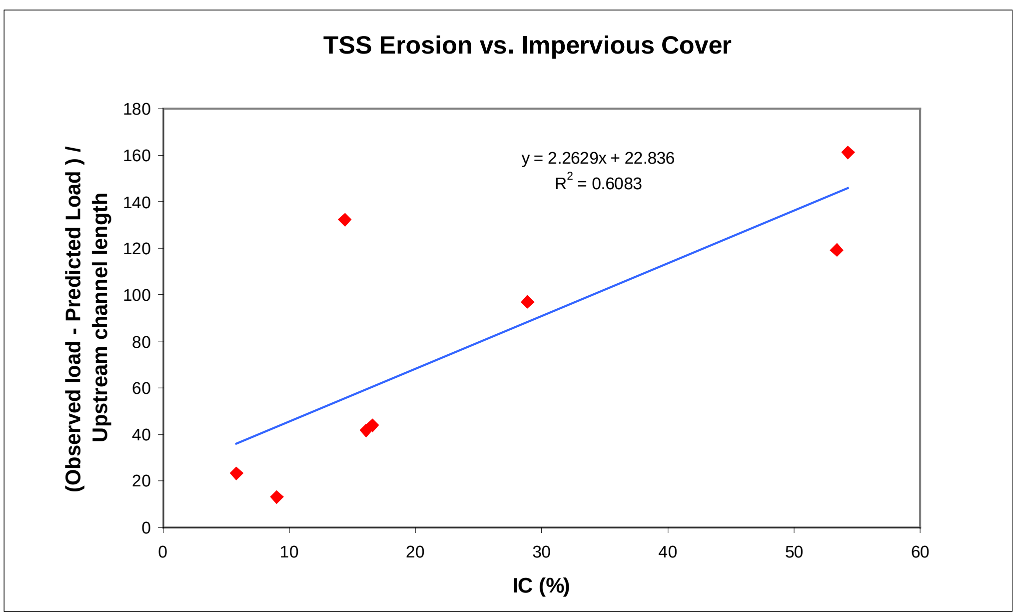


Figure 6.7: TSS erosion vs. impervious cover

The relationships between erosion and average impervious cover are summerized in table 6.6.

Table 6.6: Erosion relationships

Constituent	Equation	R^2
TSS	y = 2.2629 * IC(%) + 22.836	0.61
TN	y = 0.0033 * IC(%) + 0.0267	0.56
TOC	y = 0.0348 * IC(%) + 0.5444	0.54
TP	y = 0.0019 *IC(%) - 0.0177	0.88

6.3 CONSTRUCTION LOAD

The construction load is computed in a slightly different way. This load is generated during the construction period on all vacant land areas. The first step consists of locating these areas.

6.3.1 Zones to be developed

Newly developed areas are defined as the zones where the land use goes from undeveloped under current conditions to developed for future conditions. The percentage of area within each traffic serial zone which is going to be developed is determined as the difference between the current and the future average undeveloped land use percentage for each zone. The current undeveloped conditions can be determined by averaging the land use distribution over the cells within the various traffic serial zones. The principle is to attribute the value 1 to all the cells corresponding to the undeveloped land use and 0 to the others. The average value of the cells over the traffic serial zones gives the average undeveloped land use percentage for each zone. This value can be computed in Arc/Info with the function *zonalmean* (as shown below) or in ArcView with the script Qual.Average.

When creating the grid for the undeveloped land use (value 0 or 1), it is important to select the cells from a grid containing all land uses. If this grid is directly built from the land use coverage, one cell may be attributed several land uses. This can be verified by creating several grids corresponding to the different land uses and adding their percentage: the total values will be more than 100%, especially in small zones. This error comes from the discretization process which converts a coverage to a grid. By selecting the land uses from a grid, each cell can be only attributed the dominant land use in that cell (Figure 6.8).

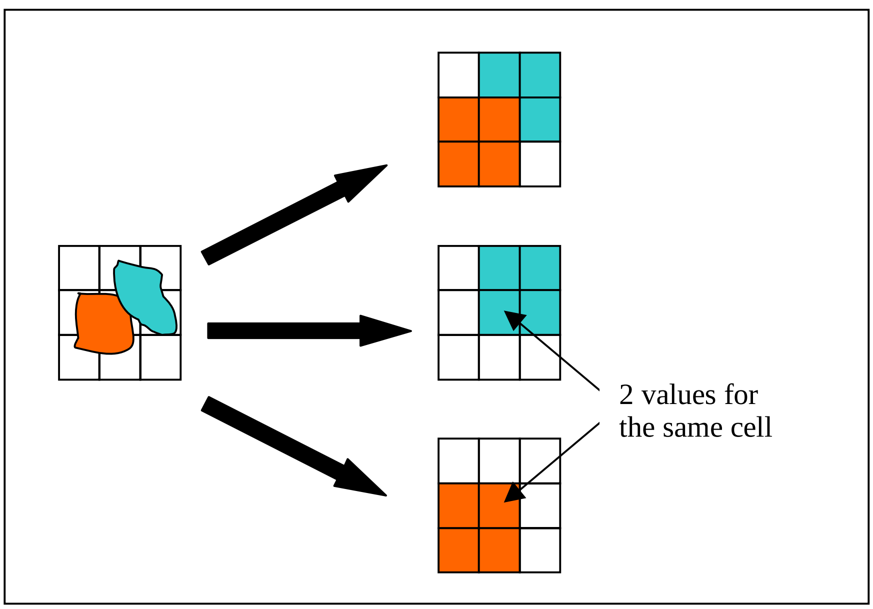


Figure 6.8: Error in conversion from vector to grid

The land use grid is based on the *newcode* field of the current land use coverage.

Grid: **luse_gr = polygrid (finluse , newcode)**

The cells corresponding to undeveloped (land use code 900) are then selected.

Grid: **undev_gr = con (luse_gr == 900 , 1 , 0)**

The cell value is finally averaged over each Traffic Serial Zone.

Grid: **undev_avg = zonalmean (zone , undev_gr)**

Any cell of this grid gives the average percentage of the area within the corresponding traffic serial zone designated as current undeveloped land use. The future undeveloped percentage is obtained directly in the table characterizing the traffic serial zones since it is one of the parameter for which predictions were made under future conditions by the City of Austin’s Planning Department. The difference between these two values gives the percentage of each traffic serial zone which is going to be developed.

The exact distribution of the part of the zones to be developed is not known. The assumption made to solve this problem is that considering that a given percentage of the total area is going to be totally developed is the same as considering that the whole area will be partially developed (Figure 6.9).

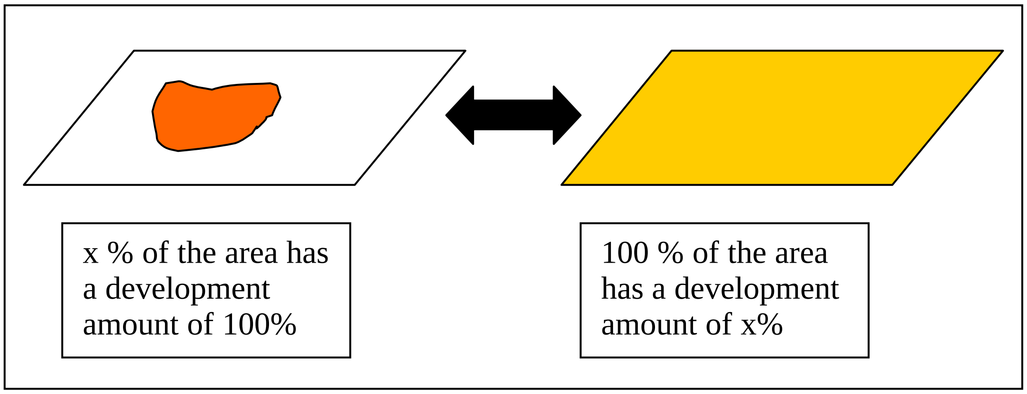


Figure 6.9: Assumption for percentage of development

A grid (*buildup*) containing the ratio of development in each cell is created.

6.3.2 CONSTRUCTION LOAD COMPUTATION

Construction load is only computed for total suspended solids (TSS). The City of Austin supplied a TSS EMC of 600mg/l, i.e. more than three times the value used in the study (190 mg/l), and an average runoff coefficient of 0.5 to use in the construction load computation. This runoff coefficient corresponds to an impervious cover of 62%. The discharge and the load produced by each cell are first defined, and then the *flowaccumulation* function allows one to compute the total construction load (*tss_b*). The units of this load are kg/number of years of construction. In this case, the construction period chosen is 45 years, hence the annual load is obtained by dividing this load by 45.

Grid: **flow_b = buildup * 0.5 * 31.08 * 5 / 189216**
Grid: **tss = flow_b * 600 * 3.048 * 3.048 * 3.048 * 86400 * 365 / 100000**
Grid: **tss_b = flowaccumulation (burn_fdr , tss)**

Table 6.7 shows a comparison between current loads and construction loads both expressed in kg/yr. Constructions are not important in urban watersheds, which are already developed. Therefore construction loads for these watersheds are less than current loads (Waller, Shoal, Boggy). On the other hand, the currently mostly undeveloped watersheds (Onion, Bear, Barton, Slaughter and Williamson) are going to be developed: construction loads are bigger than current loads for these watersheds.

Table 6.7: Comparison between current loads and construction loads for TSS

Usgs#	Name	Construction load kg/yr	Current load kg/yr	Construction load / Current load
8157500	Waller at 23rd	7.63E+02	8.35E+05	0.04
8157000	Waller at 38th	5.65E+02	4.22E+05	0.06
8156800	Shoal at W 12th	8.25E+03	1.21E+06	0.31
8156700	Shoal at NW Park	7.13E+03	7.36E+05	0.44
8158050	Boggy at US183	2.13E+04	1.40E+06	0.69
8158600	Walnut at Webberville Road	2.68E+05	4.16E+06	2.89
8154700	Bull at Loop 360	8.27E+04	1.22E+06	3.06
8158970	Williamson at Jimmy Clay Road	1.06E+05	1.23E+06	3.88
8158920	Williamson at Oak Hill	3.31E+04	3.77E+05	3.94
8158840	Slaughter at FM Road 1826	6.29E+04	4.00E+05	7.08
8155240	Barton at Lost Creek	5.02E+05	2.69E+06	8.39
8155300	Barton at Loop 360	5.28E+05	2.53E+06	9.41
8155200	Barton at SH 71	4.55E+05	2.03E+06	10.06
8158810	Bear at FM Road 1826	7.95E+04	2.74E+05	13.06
8158700	Onion at Driftwood	6.60E+05	2.11E+06	14.05
8159000	Onion at US 183	2.03E+06	5.16E+06	17.73
8158800	Onion at Buda	9.32E+05	1.41E+06	29.68

External loads have been computed based on impervious cover/event mean concentration relationships. In-stream erosion has been defined from the difference between observed loads and predicted external loads. Construction loads have also been computed.

Chapter 7 Best Management Practices

To reduce non-point source pollution loading, the City of Austin uses Best Management Practices (BMPs) such as ponds or sand filters. The objective of this project is to study the effectiveness of existing and planned BMPs. The action of the BMPs is modeled by two different ways, depending whether the BMP's location is known.

7.1 LOCATED BMPS

Located BMPs are defined by a point in a point coverage and by their effect on the load, defined either by load removed or by efficiency. It is easier to model the BMPs defined by load removed than by efficiency since, in the second case, the load removed depends on the value of the incoming load. The located BMPs in this study were defined by load removed. The second approach was also developed and is explained in section 7.1.2.

7.1.1 Effects of BMPs defined by load removed

The City of Austin staff has compiled the loads removed of different constituents at the located BMPs and has created point coverages corresponding to their location by geocoding (Figure 7.1). Two types of BMPs (residential *city1*, commercial *com1pd27*) are considered for current conditions and three types (residential *cityf1*, commercial *com1pd27* and retrofit *future1*) for future conditions. The point coverages to use for residential and commercial BMPs are the same for both conditions. The loads removed by commercial BMPs remain between current and future conditions; thus the model is run only once for this coverage. However, loads removed from residential BMPs increase between current and future conditions and the model must be run for each scenario. 361 BMPs are represented in Figure 7.1: 121 residential, 229 commercial and 11 retrofit.

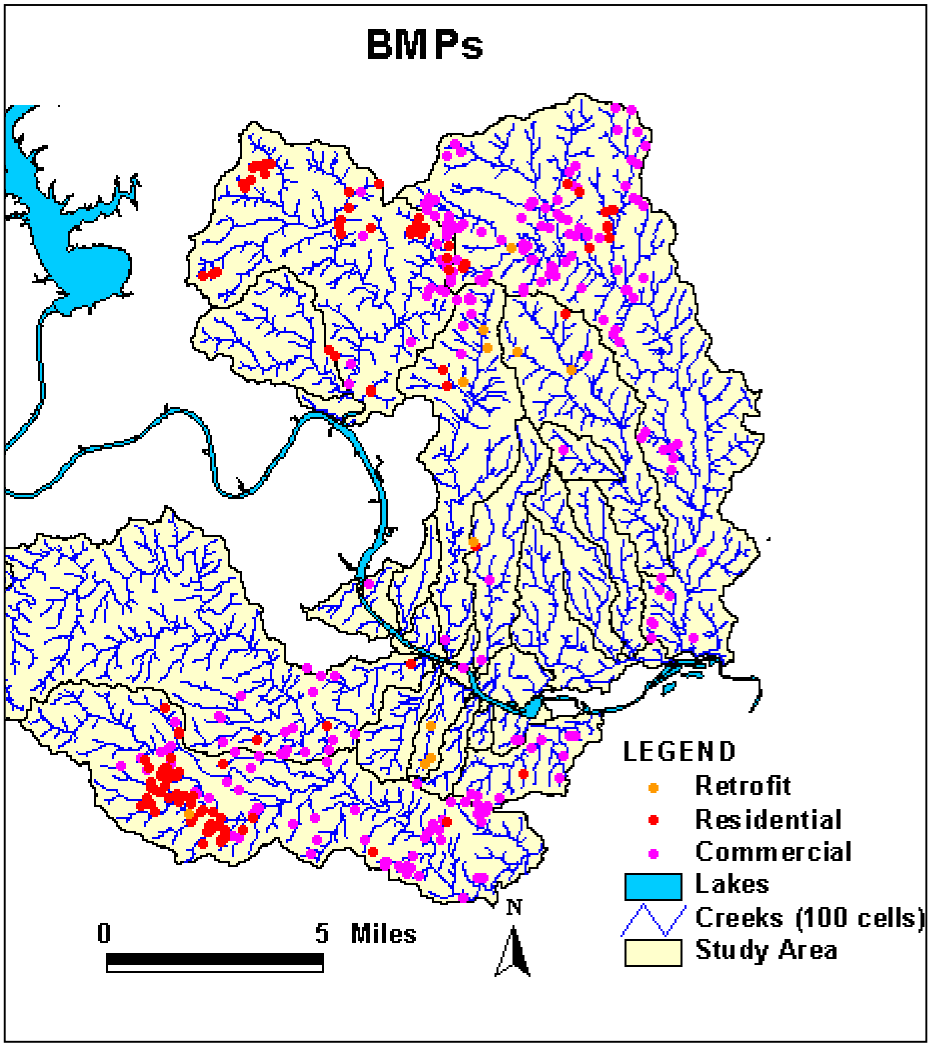


Figure 7.1: Located Best Management Practices

The total load removed by the BMPs upstream of any cell is determined by doing a weighted flowaccumulation on a grid which has for values the loads removed at the BMPs and no data elsewhere (Figure 7.2). Each type of BMP is studied separately to obtain the effect of each category. However, if the objective is to get the total load removed, the BMPs can all be modeled together by using a unique point coverage.

The City of Austin supplied tables containing the loads removed associated with each BMP category. These tables were saved as .dbf files and added in ArcView. The one to one relationship existing between the records in these tables and in the corresponding BMPs coverages attributes table allows one to join the tables together, using the script *Qual.Join* in ArcView (section 2.4.2). Once the tables have been edited, the grids corresponding to the point coverage are created so that the cell value at the BMPs is equal to the mass of constituent removed.

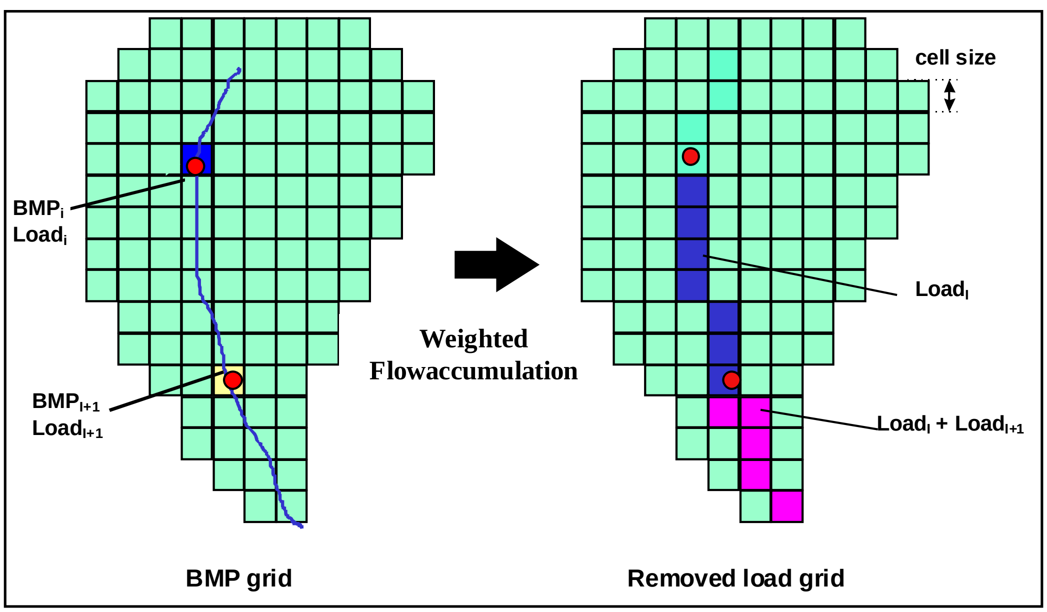


Figure 7.2: Computation for located BMPs defined by load removed

- **Analysis using Arc/Info**

For example, considering the point coverage *city1* (residential for current conditions) and the field *tss*, which indicate the TSS load removed, the procedure is:

Create a grid point coverage
Grid: **tss_pt = pointgrid (city1 , tss)**

Sum the loads removed
Grid: **tss_cc = flowaccumulation (burn_fdr , tss_pt)**

Since the flowaccumulation function does not handle negative values, if the grid contains both negative and positive values (e.g. for NO_3), two positive grids must be created. The first one contains the cells with positive values. The second one contains the absolute values of the cells with negative values. A *flowaccumulation* is done for each grid and the second grid is subtracted from the first one.

Grid: **no3_pt = pointgrid (city1 , no3)**
Grid: **no3_ptp = con (no3_pt > 0 , no3_pt)**
Grid: **no3_ptn = con (no3_pt <= 0 , - no3_pt**)
Grid**: no3_ccp = flowaccumulation (burn_fdr , no3_ptp)**
Grid: **no3_ccn = flowaccumulation (burn_fdr , no3_ptp)**
Grid: **no3_cc = no3_ccp - no3_ccn**

- **Analysis using ArcView**

The BMPs can also be modeled in ArcView. The script *Qual.BMPload* (section 8.4.1 and Appendix C) used to model the BMPs is also based on the principle developed in this section.

7.1.2 Effects of BMPs defined by efficiency

BMPs are usually defined by their removal efficiency, i.e. the percentage of the incoming load which is removed. Although the BMPs used in this study are defined by the mass removed, a procedure in the case where they are defined by the removal efficiency is created. This approach is more complicated because the mass of load removed now depends on both the BMP efficiency and the incoming load.

New load =
- Initial load *upstream of the BMP*
- (Initial load – Removed load) *downstream of the BMP*

where Removed load = Removal efficiency * Initial load at the BMP
$= eff_{BMP} * load_{BMP}$

For one BMP, the principle also uses the *flowaccumulation* function to create a removed load grid. For cells that have loads removed, the value for that cell is the actual load removed (Figure 7.3).

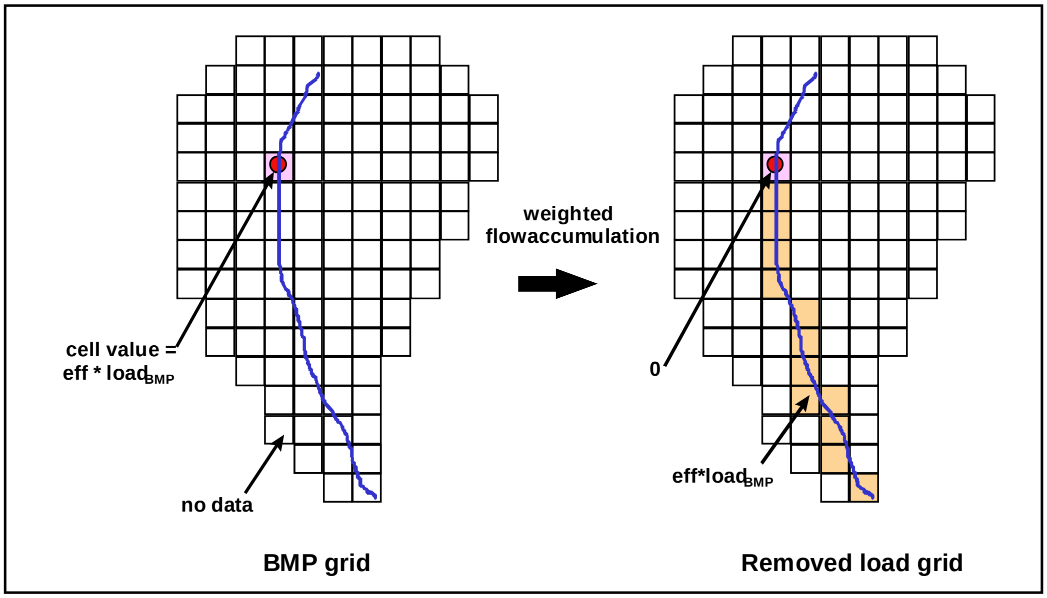


Figure 7.3: Removal load grid for located BMPs defined by efficiency

This removed load grid is subtracted from the initial load grid to create the new load grid after load removal at the BMPs. In Figure 7.4, the lowest loads are represented in blue and the highest in red.

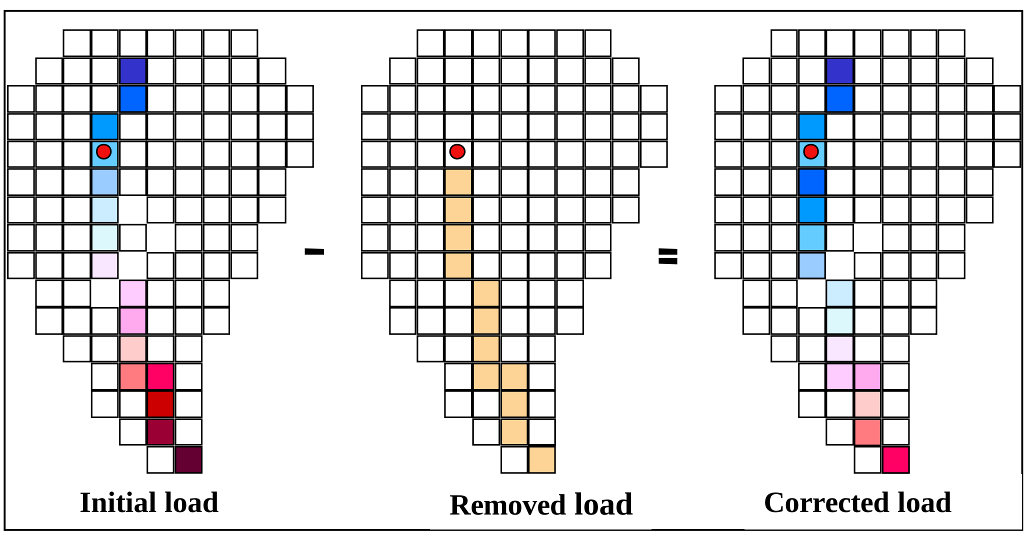


Figure 7.4: Corrected load

In the case of nested BMPs, the load removed by the most downstream BMP (BMP_2) is affected by the load removed by the BMP upstream (BMP_1) (Figure 7.5).

$$\textbf{Load removed by } \mathbf{BMP_2 = eff_2 * [load_2 - eff_1*(load_1)]} \qquad [7.1]$$

where load1 is the initial load at BMP_1 and $load_2$ the initial load at BMP_2.

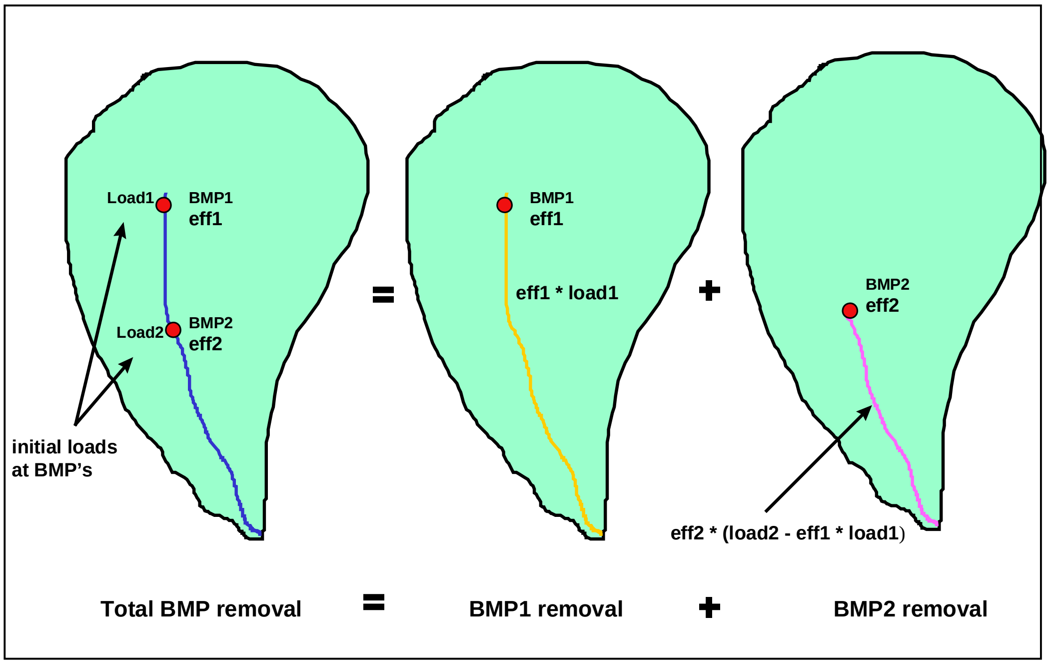


Figure 7.5: Total BMPs removal

If BMP_n is located downstream of n-1 BMPs of efficiencies eff_1,..., eff_{n-1}, the load removed by BMP_n depends on the incoming load to the BMP_n, load which has been treated by n-1 upstream BMPs.

The general relationship is:

$$\textbf{Load removed by } \mathbf{BMP_n = eff_n*(load_n-eff_{n-1}*(load_{n-1}-eff_{n-2}*(load_{n-2}-\ldots)\ldots eff_2*(load_2-eff_1*load_1)\ldots)} \qquad [7.2]$$

The existence of nested BMPs is determined by doing a weighted *flowaccumulation*. The weight grid is a grid with positive values at the BMPs and no data elsewhere. Since *flowaccumulation* adds the value of the cells located upstream of the cell considered, the value in the new grid is zero for non-nested BMPs and positive for nested BMPs. The loads removed is first computed for the non-nested (more upstream) BMPs, which are given the order 0. The other BMPs are located downstream of at least another BMP and are assigned an order based on the flowaccumulation value at the BMP. Comparing two BMPs, the downstream BMP has a larger flowaccumulation value since this value represents the number of cells located upstream of the cell considered. The watershed of the downstream BMP includes the watershed of the upstream BMP (Figure 7.6). The maximum order corresponds to the number of nested BMPs. For example, the watershed on the left in Figure 7.7 contains 4 non-nested BMPs and 5 nested BMPs. The maximum order is 5.

The load removed is computed separately for each nested watershed after computing the load removed by non-nested BMPs. This is done by setting the efficiency of the BMPs whose order is different from the order considered to zero. For example, if the order is 1, all the BMPs whose order is different from 1 have a zero removal efficiency, i.e. only the removal efficiency of the BMP with order 1 is considered. The number of flowaccumulations needed to compute the load removed is equal to the number of nested BMPs plus one (the computations for non-nested BMPs are all done at the same time).

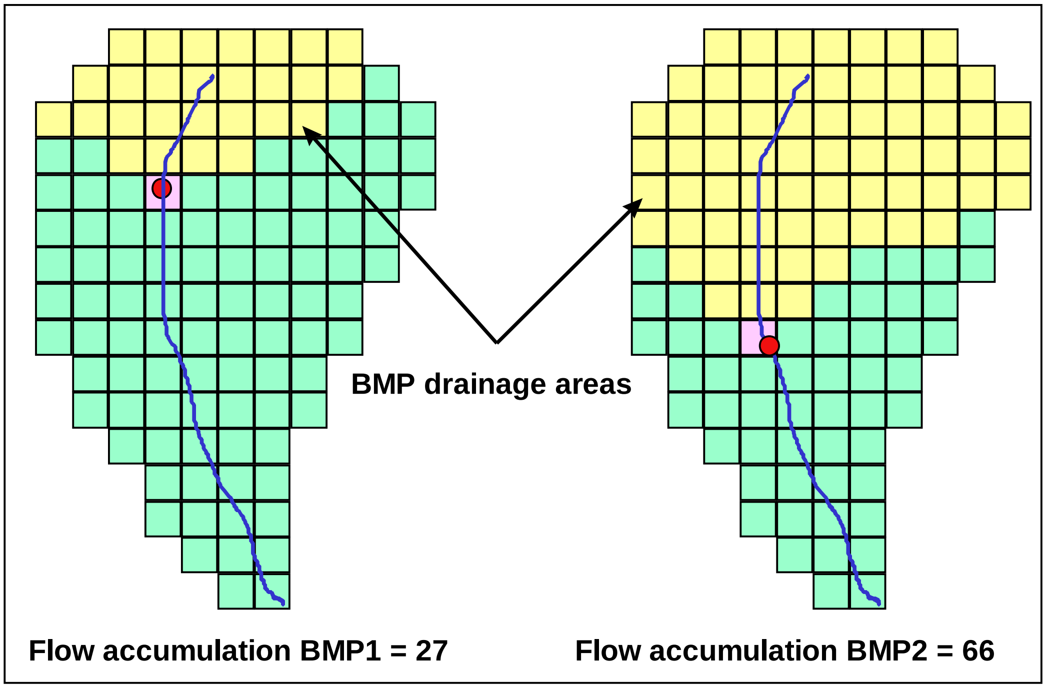


Figure 7.6: BMP classification

The *flowaccumulation* function is precisely the execution time limiting function of the program. To decrease the execution time, the number of iterations must be reduced. This can be implemented by modeling each watershed separately. Two independent watersheds can not have nesting problems and their BMPs can be ordered totally separately. Figure 7.7 shows the advantage of the method, which in the simple case of two watersheds, enables one to reduce the number of iterations from 9 to 6. The advantage will be all the more important when the system being considered is complex.

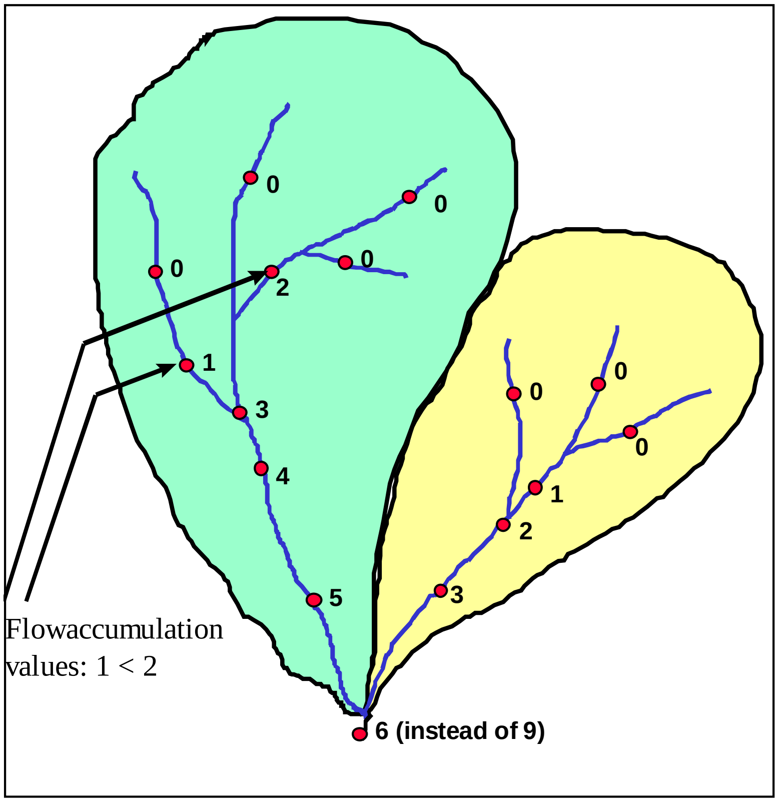


Figure 7.7: Classification for non-nested watersheds

The ranking method is programmed in the scripts *Qual.BMPeff* and *Qual.BMPcomp*, which is a subroutine of *Qual.BMPeff* (Chapter8 and Appendix C).

The inconvenience with this procedure is that the BMPs have small drainage areas. It is important that the Digital Elevation Models used in the study give a very accurate description of the topography. A comparison between the data gathered in the field and the BMPs drainage areas determined by delineation shows the 30m cell used in the study are not accurate enough. However, the use of new sources of data such as the digital orthophotos with an increased accuracy should enable this approach.

If observed drainage areas are available, the solution consists in correcting the efficiency in function of the errors between observed and computed drainage areas. The computed drainage areas are obtained with the flowaccumulation grid: the flowaccumulation value indicates the number of cells located upstream of a given cell, i.e. the number of cells in the drainage area of that cell. For example, if the observed drainage areas are given in acres, the relationship between computed areas in acres and flowaccumulation value is, for the cell area in square feet:

$$\text{Computed area}\,[\text{acre}] = \frac{\text{flowaccumulation value} * \text{cell area}\,[\text{ft}^2]}{43560\,[\text{ft}^2\,/\,\text{acre}]} \qquad [7.3]$$

A coefficient defining the relationship between observed and computed drainage areas for each BMP can be defined (equation 7.4).

$$\text{correction}_{\text{area}} = \frac{\text{observed drainage area}}{\text{computed drainage area}} \qquad [7.4]$$

A BMP treats the loads generated in its drainage area with a given removal efficiency, which must be corrected to be applied to the drainage area computed by the model. Since the drainage areas for the BMPs are small, the difference between the loads generated on the observed and on the computed drainage area is assumed to be proportional to the difference in areas. The load effectively removed at the BMPs can be obtained by using corrected efficiencies (equation 7.5).

$$eff_{corr} = eff * correction_{area} \qquad [7.5]$$

If the observed area is larger than the computed one, the area correction coefficient is bigger than 1 and the corrected efficiency is bigger than the initial efficiency. The method presented in this section applied to the BMPs coverage defined with corrected efficiencies.

7.2 NON LOCATED BMPS

In the future, new BMPs will be associated with each new construction. Since it is not possible to accurately determine with great accuracy the areas which will be developed, it is difficult if not impossible to know the precise locations of any future BMP. The approach used for the located BMPs is not valid and the average effect of BMPs over a given area must be computed instead.

The planned BMPs treat newly developed areas, which are defined as the zones where the land use changes from undeveloped under current conditions to developed under future conditions. It is impossible to precisely locate these areas, therefore the average development occurring in each traffic zone is used instead (*buildup*, section 6.3.1).

7.2.1 BMP zones

The non-located BMPs are assumed to remove the load directly in the cell where it is generated. Direct runoff load computation has already been explained in Chapter 5.

The other variable needed to compute the removed load in each cell is the associated BMP removal efficiency.

Different BMPs are used to treat the newly developed zones: the combination of BMPs are based on the characteristics (recharge zone…) and on the location (City of Austin, eastern suburban…) of the watersheds (Table 7.1). Eight zones are defined. A combination of the five following types of BMP are applied in each zone:

- SED1: sediment pond with ½” water quality volume, “on-line” or “off-line” system.
- SAND2: “on-line” sand filtration system with no pretreatment and ½” water quality volume.
- SAND3: “off-line” sand filtration system with pretreatment and ½” water quality volume.
- COMP: City of Austin’s Composite Ordinance
- SOS: City of Austin’s Save Our Spring Ordinance

Table 7.1: Percentage of BMPs per BMP zones (source: COA=City of Austin)

	Barton Springs Zone[1]				Non-Barton Springs zone			
	Recharge Zone		Contributing Zone		COA Jurisdiction			Non-COA
BMP	COA	Non-COA	COA	Non-COA	Urban[2]	Eastern Suburban[3]	Western Suburban[4]	
ZONE	1	2	3	4	5	6	7	8
NONE				100%	73%	14%		100%
SED1			10%					
SAND2	44%	100%	34%		17%	24%	44%	
SAND3					10%	62%	56%	
COMP	36%		36%					
SOS	20%		20%					

[1] Barton Springs Zone watersheds include Barton, Williamson, Onion, Bear, Little Bear and Slaughter creeks in recharge and contributing zones.

[2] Urban watersheds are Shoal, Waller, East Bouldin, West Bouldin, Blunn, Fort, Tannehill, Buttermilk, Little Walnut, Harper's Branch, Johnson, Boggy, Town Lake adjacent and the segment of Barton below Barton Springs road, i.e. the very end of the watershed (administratively, the city treats this area as an urban watershed). For Shoal creek only, an urban watershed, apply the SAND2 BMP to 100% of new development within the northern Edwards recharge zone to reflect the state's Edwards rule. This is different from the "urban scenario" and applies to that portion of Shoal in the recharge zone.

[3] Eastern Suburban watersheds are Walnut, Country Club, and Williamson, Bear, Little Bear and Slaughter downstream of the recharge zone (and all remaining watersheds in the eastern part of the city).

[4] Western suburban watershed are Bull, Eanes, Lake Austin, Bee, Little Bee, Taylor Slough North, Taylor Slough South, Huck Slough and Dry Creek.

This classification is based on the following assumptions:

- Non-COA Barton Springs Contributing Zone and Non-Barton Springs watersheds are assumed to have no water quality protection regulations (i.e., other cities and Water Quality Protection Zones provide little or no protection in these areas).

- In Western suburban watersheds, assume that all future projects will have some control (Lake Austin ordinances (1980/81)). There are no sedimentation controls because these were not typically allowed as a form of treatment.

- Assumption for Barton Springs zone current conditions is the same for SAND2.

- Zones outside of the city jurisdiction but in the Edwards recharge zone (north or south), apply the SAND2 BMP.

The recharge and contributing zones are obtained by an Arc/Info *union* function with the watersheds, the Edwards Aquifer (*Edwards*) and the City jurisdiction (*juris*, Figure 7.8*)* coverages.

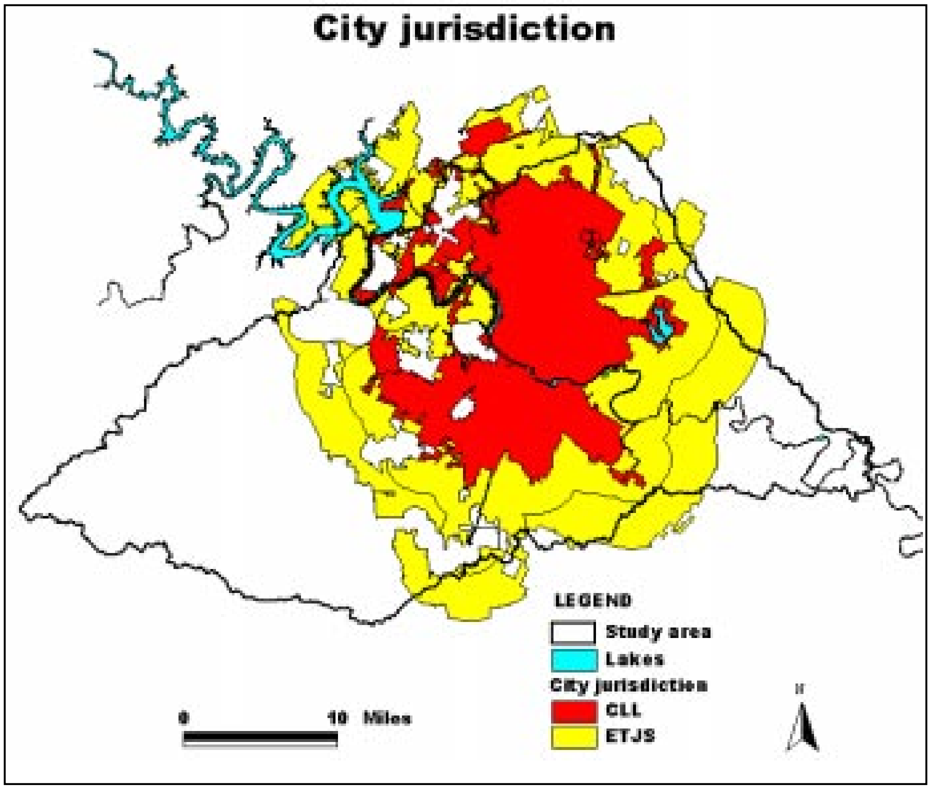


Figure 7.8: City jurisdiction

*   CLL: City Limit Line
** ETJS: Extra territorial Jurisdictions

The CLL (City Limit Line) and ETJs (extra territorial jurisdictions) are within the City's jurisdiction. The watersheds are characterized as urban (1), eastern (2) or western suburban (3) in a new field created in ArcView. A coverage with zones fitting the criteria in Table 7.1 is created in ArcView (*futbmp_cv*) by using the selection tool and are assigned a zone number (in field *bmpzone*). A grid corresponding to the BMP zones is then created (Figure 7.9).

Grid: **futbmp_gr = polygrid (futbmp_cv , bmpzone)**

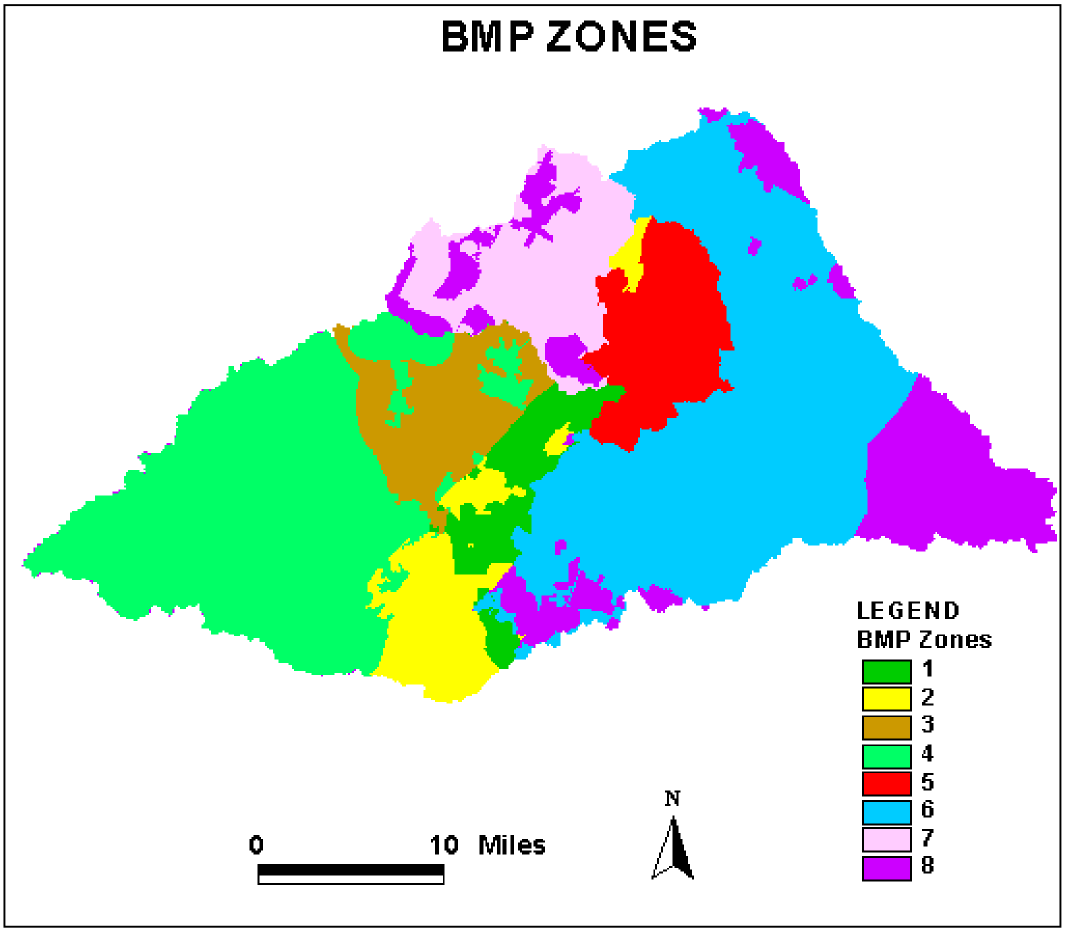


Figure 7.9: Future conditions BMP zones

7.2.2 Removal efficiencies

The BMPs treat only direct runoff. Their effective removal efficiency is a function of the percentage of the volume of water that they treat (annual capture volume, ACV) and of their average event removal efficiency (AERE).

$$\mathbf{Eff = ACV * AERE} \qquad [7.6]$$

- **Capture volume**

A distinction must be made between on-line and off-line BMPs. Off-line BMPs capture only a certain amount of runoff: the flow greater than the design flow by-passes the system. The average amount of runoff captured varies with the impervious cover and the BMP water quality volume (capture volume). For on-line BMPs (e.g. wet ponds), all runoff enters the BMP and no capture volume analysis is necessary.

Some on-line sedimentation (SED1) and sand filtration systems are treated as off-line using a conservative approach which is justified since on-line systems perform poorly when the runoff volume is greater than the water quality volume.

For off-line systems, the capture volume is estimated on an annual average basis as an annual capture volume (ACV) which varies with regulatory requirements:

- Ordinances require a ½” water quality volume regardless of site impervious cover.
- More recent ordinances increase the water quality volume as impervious cover increases.

Linear relationships relating average capture volume to impervious cover have been established by the City of Austin (Table 7.2).

Table 7.2: Capture volume/Impervious Cover relationships (0<IC<1)

BMP	ACV (0<ACV<1)
SED1 , SED2 (½”)	0.996 – 0.4714*IC
SAND3, COMP	0.9762 – 0.154*IC
SOS	1

These relationships show that the percentage of treated direct runoff decreases when the impervious cover increases since more direct runoff is generated. The capture volume grids for the different BMPs can be created from the future impervious cover grid (*futic_gr*) by applying the equations shown in Table 7.2.

Grid: **acv1 = 0.996 - 0.4714 * futic_gr**
Grid: **acv2 = 0.9762 - 0.154 * futic_gr**

- **Average event removal efficiency**

Over its life, a BMP will achieve 80% (SED1) to 90% (non-degradation BMPs) of its projected performance. The average event removal efficiency (Table 7.3) which indicates the load efficiency removal for each constituent takes this factor into account.

Table 7.3: SED1, SAND2, SAND3 average event removal efficiencies

Average Event Removal Efficiency (%)		
Constituent	SED1	SAND2 SAND3
TSS	40	68
BOD	24	40
COD	24	52
TOC	16	48
NO3	0	-25
NH3	8	48
TKN	16	40
TN	16	24
DP	0	0
TP	20	48
Cu	16	40
Pb	32	64
Zn	16	40

Composite ordinance (COMP) and Save Our Springs ordinances (SOS) have a 81% removal efficiency.

- **Removal efficiencies in the BMP zones**

Grids with the removal efficiency associated to each BMPs zone for each constituent can be created in Arc/Info by taking the product of the average event removal efficiency by the associated capture volume for each BMP and by weighting the contribution of each BMP type within the different zones shown in the grid *futbmp_gr*.

Grid: **eff1 = con (futbmp_gr == 1 , 0.44 * acv1 * SAND2 + 0.36 * acv2 * COMP + 0.2 * SOS , futbmp_gr)**

Grid: **eff2 = con (futbmp_gr == 2 , acv1 * SAND2, eff1)**

Grid: **eff3 = con (futbmp_gr == 3 , (0.1 * SED1 + 0.34 * SAND2) * acv1 + 0.36 * COMP * acv2 + 0.2 * SOS , eff2)**

Grid: **eff4 = con (futbmp_gr == 4 or futbmp_gr == 8 , 0 , eff3)**

Grid: **eff5 = con (futbmp_gr == 5 , 0.17 * SAND2 * acv1 + 0.1 * SAND3 * acv2 , eff4)**

Grid: **eff6 = con (futbmp_gr == 6 , 0.24 * SAND2 * acv1 + 0.62 * SAND3 * acv2 , bodeff5)**

Grid: **eff0 = con (futbmp_gr == 7 , 0.44 * SAND2 * acv1 + 0.56 * SAND3 * acv2 , eff6)**

7.2.3 Load corrected by the BMPs

- **Removed load**

The load treated is the load generated in the newly developed areas. The removal efficiency in each cell (*eff0*) must be multiplied by the development rate in that cell (given by the grid *buildup*, section 6.3.1).

Grid: **eff = buildup * eff0**

The resulting efficiency grid (*eff*) is multiplied by the direct runoff load generated by each cell (*loadcell*) to yield the load removed in each cell. A flowaccumulation produces then the total load removed at any location in the grid (*load0rem*).

Grid: **loadremcell = loadcell * eff**
Grid: **load0rem = flowaccumulation (burn_fdr , loadremcell)**

If no recharge zone is taken into account, the new load is obtained by simply subtracting the removed load from the load previously computed (*load*).

Grid: **newload = load - load0rem**

The method is much more complicated with a recharge zone since the BMPs modify the in-stream concentrations in two ways:

1. Load removal
2. Direct runoff removal for non discharge BMPs (COMP and SOS).

Base flow and direct runoff are assumed to be well mixed in the creeks so that direct runoff and base flow lost in the recharge zone have the same concentration (*co*) as the total flow at the location where the losses occur. Since the concentration changes and the recharge stays constant, the load lost to the recharge zone is also modified and must be reevaluated. Note that the in-stream concentration modifications caused by the located BMPs were assumed to be negligible and their effect on the load lost to the recharge zone was not considered. This is because the calibration was done by neglecting the effect of the current located BMPs. The impact of the located BMPs on the recharge zone could however also be implemented if necessary.

- **Flow after non discharge BMPs correction**

The direct runoff removed by the non discharge BMPs (COMP and SOS) is obtained by computing first the flow removal efficiency in the zones where these BMPs are located (zones 1 and 3), and by multiplying the efficiency grid by the direct runoff generated in each cell. A flowaccumulation will eventually create a grid containing the direct runoff retained in the non-discharge BMPs (Procedure 7.1). Without taking into account the recharge, the new flow (*totalflow1*) considering the effect of non discharge BMPs is obtained by subtracting the removed flow from the previous flow value (*totalflow0*).

Procedure 7.1: Flow corrected with non-discharge BMPs

Direct runoff coefficients (apply IC/direct runoff coefficient relationship)
Grid: **runcoef = 0.3428 * icfut_gr * icfut_gr + 0.5677 * icfut_gr + 0.0125**

Development rate in zones with non-discharge BMPs
Grid: **zone13 = con (futbmp_gr == 1 or futbmp_gr == 3 , buildup)**

Removal flow efficiency
Grid: **capt13 = zone13 * (0.36 * ACV2 * 0.9 + 0.2 * 0.9)**

Direct runoff captured
Grid: **runcell_cpt = runcell * capt13**
Grid: **runcell_cpt0 = flowaccumulation (burn_fdr , runcell_cpt)**

New total flow
Grid: **totalflow01 = totalflow0 - runcell_cpt0**

- **Load after BMPs effect**

The load without considering a recharge zone is computed by subtracting the removed load from the load without recharge previously computed (*load0*, section 6.1.2).

Grid: **load01 = load0 – load0rem** *(with load0 = co*totalflow0)*

The BMPs corrected in-stream concentration *co1* is obtained by dividing the BMPs corrected load *load01* by the non discharge BMPs corrected flow *totalflow01*.

Grid: **co1 = load01/totalflow01**

The BMPs corrected load lost to the recharge zone is computed by multiplying the corrected concentration grid (*co1*) by the grid containing the recharge volume lost in each cell (lcorr_rech), and by doing a flowaccumulation.

Grid: **loadcellrech = lcorr_rech * co1**
Grid: **rechload = flowaccumulation (burn_fdr , loadcellrech)**

The final corrected load is obtained by subtracting the load lost in the recharge zone from the BMPs corrected load without considering the recharge *(load01*).

Grid: **newload = load01 – rechload**

- **AML computation**

Procedure 7.2 is an AML which computes the corrected load if the effect of non discharge BMPs on the flow has been previously computed (i.e. *totalflow01*). An Avenue script (*Qual.BMPfut*, section 8.4.3) enables the user to compute both the corrected flow and load.

Procedure 7.2: BMPs corrected load

```
/*----------------------------------------
/*--- CORRECTEDLOAD.AML---
*-----------------------------------------
/*FUNCTION: compute the BMPs corrected load.
/*INPUTS: impervious cover grid (icfut_gr), IC/EMC relationship for direct runoff, traffic serial zones (zone_gr), direct runoff generated in each cell (runcell), buildup grid (buildup), removal efficiency grid (e.g bodeff), flowdirection grid (burn_fdr), concentration grid (co), total flow without recharge (totalflow0), total flow without recharge after effects of non discharge BMPs (totalflow01), recharge lost in each cell (lcorr_rech).
/*----------------------------------------
/*BEGIN AML EXECUTION
grid
setcell 100

/*Direct runoff EMC grid
emc_gr0 = 14 * icfut_gr + 3.5
emc_gr = con ( zone_gr == 999 , 0 , emc_gr0 )
kill emc_gr0 all

/*Load removed by the BMPs (without considering recharge)
loadcell = runcell * emc_gr * 3.048 * 3.048 * 3.048 * 86400 * 365 / 1000000
kill emc_gr all
eff = buildup * bodeff
loadremcell = loadcell * eff
kill loadcell all
kill eff all
load0rem = flowaccumulation(burn_fdr , loadremcell )
kill loadremell all
load01= co*totalflow0 - load0rem
kill load0rem all
co1=load01/totaflow01

/*Load lost in the recharge zone
loadcellrech = lcorr_rech*co1
kill co1 all
rechload = flowaccumulation (burn_fdr,  loadcellrech)
kill loadcellrech all
kill co1 all

/*New load after BMPs
newload = load01 - rechload
```

The results would however be different if the locations of the BMPs were known, since the relationship between average impervious cover and load removed is non-linear. Direct runoff is related to impervious cover by a second order equation while EMC and capture volume follow a linear relationship. Hence the average value at a location is not the same as the sum of average values.

7.2.4 Effect of non-discharge BMPs on channel erosion

Since the COMP and SOS BMPs applied in the City jurisdiction within the recharge zone are no-discharge BMPs, they decrease the volume of direct runoff in the streams. Direct runoff causes channel erosion in the streams: since its volume decreases, the channel erosion also decreases.

Previous studies conducted in section 6.2 have shown that erosion could be linearly related to the average impervious cover, which is related to the amount of direct runoff in the stream. The decreased direct runoff due to the ordinance BMPs can be translated at the cell level by a decreased runoff coefficient, and hence a decreased impervious cover, by applying the impervious cover/runoff coefficient relationship for direct runoff (Figure 7.10).

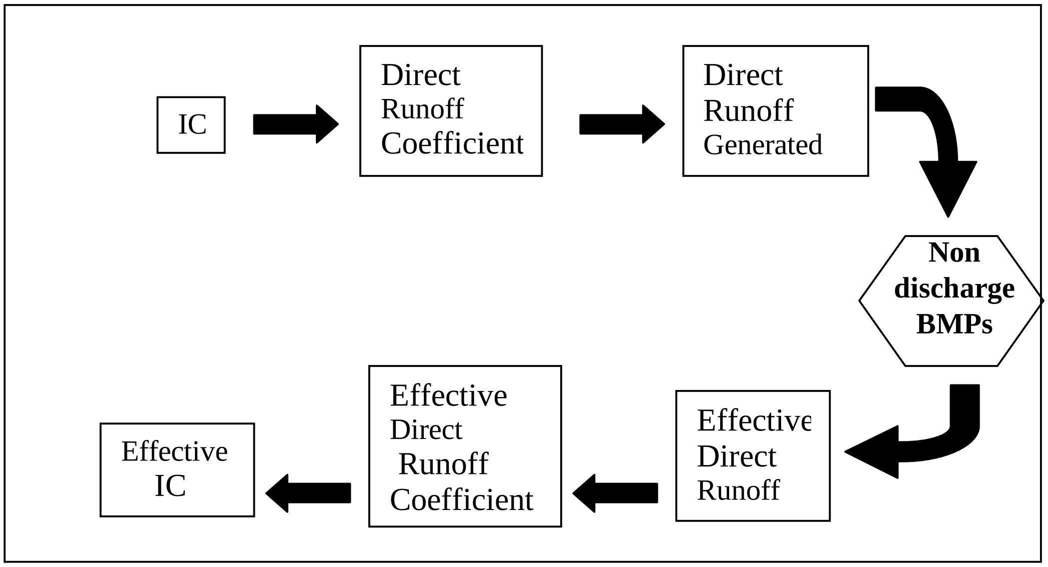


Figure 7.10: Effective impervious cover

The equation relating direct runoff coefficients to impervious cover is:

$$\text{Runcoef} = 0.3428 * \text{IC}^2 + 0.5677 * \text{IC} + 0.0125, \text{ with } 0<\text{IC}<1 \qquad [7.7]$$

Given the value of the direct runoff coefficient (runcoef), the impervious cover can be obtained by looking for the positive root of this second order equation (eq. 7.8).

$$\text{IC} = \frac{-0.5677 + \sqrt{0.5677^2 + 4 * 0.3428 * (\text{Runcoef} - 0.0125)}}{2 * 0.3428} \qquad [7.8]$$

The value of the effective runoff coefficient is obtained in Grid.

Grid: **eff_runcoef = runcoef * (1 - capt13)**
Grid: **xx1 = (0.5677)*(0.5677) + 4 * 0.3428 * (eff_runcoef - 0.0125)**
Grid: **xx2 = sqrt (xx1)**
Grid: **xx3 = -0.5677 + xx2**
Grid: **eff_futic1 = xx3 / 2 / 0.3428**
Grid: **efficfut = merge (eff_futic , icfut_gr)**

The average effective impervious cover is then obtained by doing a flowaccumulation.

Grid: **efficfut_fac = flowaccumulation(burn_fdr , efficfut)**
Grid: **efficfut_avg1 = efficfut_fac / burn_fac**
Grid: **efficfut_avg = merge (efficfut_avg , efficfut)**

The effective impervious cover grid is then used to compute the new erosion coefficients.

Best Managements Practices have been modeled in three different ways:

- Located BMPs defined by load removed.
- Located BMPs defined by efficiency.
- Non located BMPs defined by efficiency.

The effects of non-discharge BMPs (COMP and SOS) on discharge and on erosion (effective impervious cover concept) have also been modeled.

Chapter 8: The ArcView Model

During this study, Avenue scripts were created to automate several water quality modeling functions. The water quality model was also converted from Arc/Info to Avenue. The codes of the scripts are in Appendix C.

To make the model user-friendly, these scripts were customized with menus and buttons in a project and an ArcView Extension (*Qual.avx)* based on that project was created. The following sections describe the extension, indicate how it was created and how it can be modified, and show in detail how to use the scripts for computing flows, loads and BMP effects.

The objective of this chapter is not to explain the methodology: that has been done in the previous chapters.

8.1 ARCVIEW NON-POINT SOURCE POLLUTION EXTENSION

8.1.1 Extension installation

If the menus *Qual* and *QualTools* are not in the menu bar when a View window is active, then the Water Quality extension needs to be installed.

The Water quality extension, like any other ArcView extension, is installed by using in the menus the command *File/Extensions* (Figure 8.1). This command is accessible when the Project window (The Window with icons representing Views, Tables, Charts, Layouts and Scripts) is active. By default, the Project window is active in a new project.

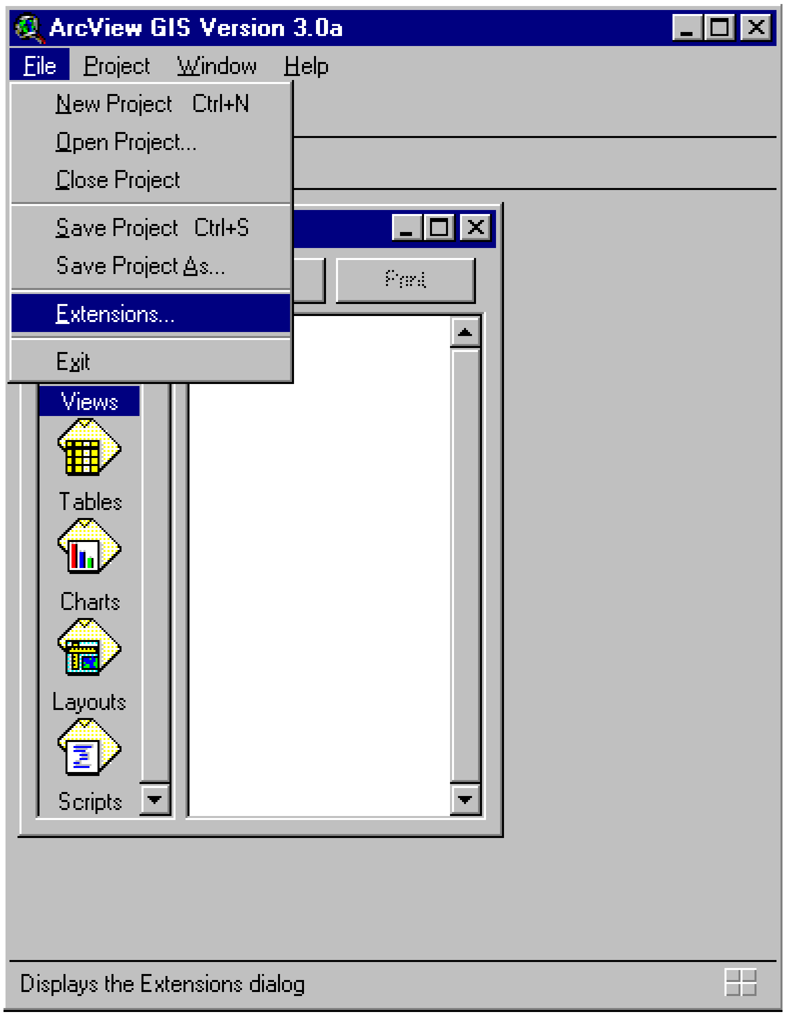


Figure 8.1: Command to install the extension

The Extensions window appears (Figure 8.2). Note that it is necessary to select also the Spatial Analyst extension to be able to use the Water Quality extension: a check mark must be put in both boxes. The Spatial Analyst extension can be obtained from ESRI. The Water Quality extension (*qual.avx*) is the extension created in this study. This extension is in the CDROM associated with this report (Appendix B). Updated versions may be obtained from CRWR. The extension file *qual.avx* must be copied to the directory esri\av_gis30\ArcView\ext32 in order for it to appear in this list of available extensions. If the user does not have permission to write to that directory, copying the file to the c:\temp works also sometimes.

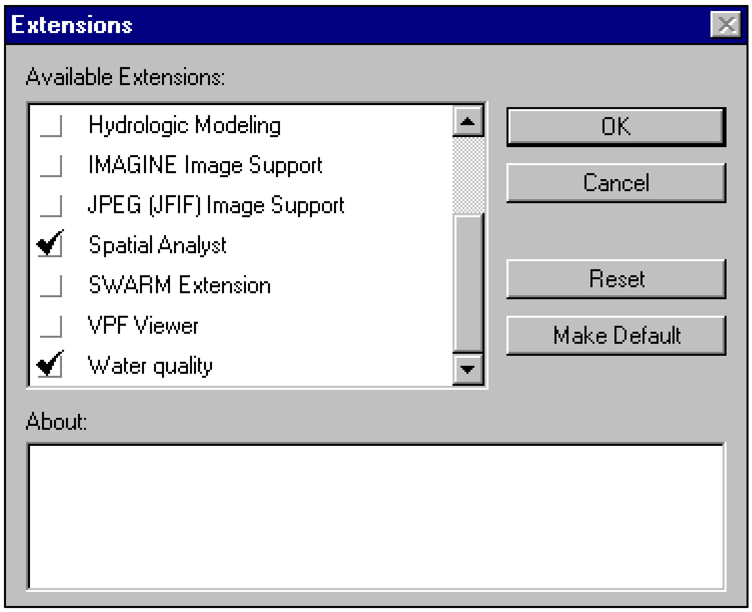


Figure 8.2: Extensions window

8.1.2 Extension description

The Water Quality extension has been installed. New buttons, menus and a new tool can now be seen. The names, functions and customization of the scripts are shown in Table 8.1. The scripts are based either on a View or on one or several Tables. If they are based on View, they can only be run when the View is active. If they are based on Table, they can only be run when the Table is active. Similarly, buttons, menus and tools are visible when either a View or a Table is active, depending on the script they are associated with.

- **View based**

When a View is active, 13 buttons, 1 tool and 2 menus from the Water Quality extension are visible (Figure 8.3).

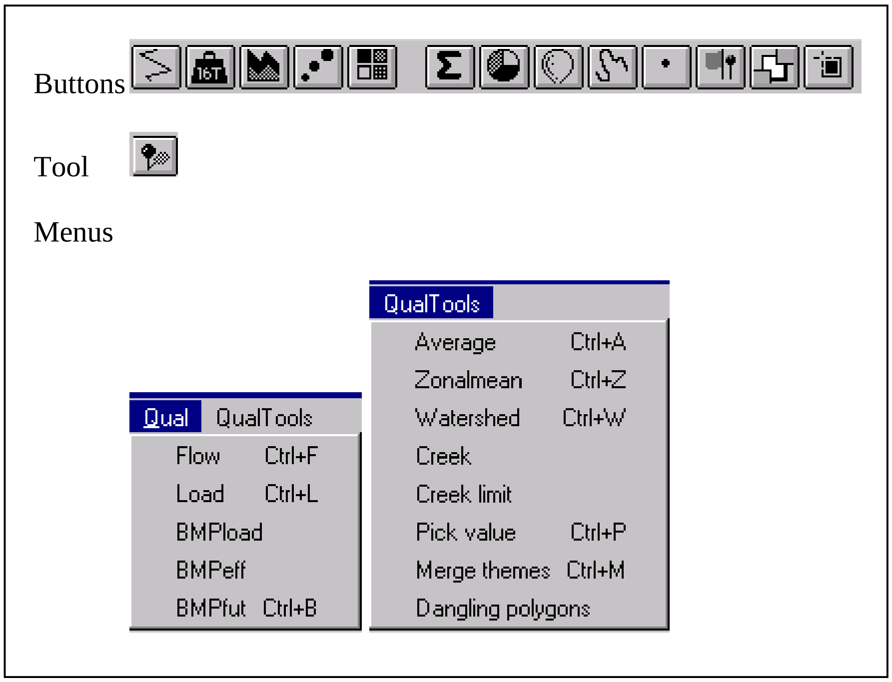


Figure 8.3: View based buttons, tool and menus

The first 5 buttons, associated with the menu *Qual*, are used for flows, loads, and BMP effects computation. Their funtion and the input files needed for them are described in detail in sections 8.2, 8.3 and 8.4. The last 8 buttons, associated with the menu *QualTools*, correspond to different utility tools used during the study, and have been described in Chapters 2, 3 and 4.

- **Table operations**

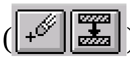
When a table is active, two buttons () and a menu (*QualTable*) appear. The first button is used to delete several fields in a table. It corresponds to the first item in the menu (*Delete fields*). The second button, corresponding to the second item in the menu (*Join permanently*), is used to join permanently two tables (Chapter 2).

Table 8.1: Description of Scripts

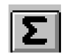
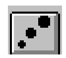
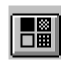
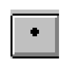
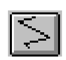
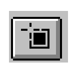
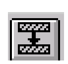
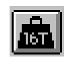
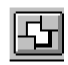
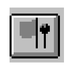
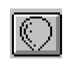
Script name	Menu	Icon	Function
Qual.Addpoint	Qual/Addpoint		Creates or adds a point to a point coverage.
Qual.Average	QualTools/Average		Computes the average based on drainage area.
Qual.BMPeff	Qual/BMPeff		Computes the load removed by located BMPs defined by removal efficiency.
Qual.BMPfut	Qual/BMPfut		Computes the new load after the effect of non located BMPs defined by removal efficiency.
Qual.BMPload	Qual/BMPload		Computes the load removed by located BMPs defined by load removed.
Qual.Creek	QualTools/Creek		Delineates creeks for a flowaccumulation threshold.
Qual.Creeklimit	QualTools/Creek limit		Creates gridpoints representing the upstream limit of the creeks for a flowaccumulation threshold.
Qual.Delete	QualTable/Delete fields		Deletes several fields in a table.
Qual.Flow	Qual/Flow		Computes the discharge (cfs).
Qual.HydroZdlsv	QualTools/Dangling		Dangling polygons.
Qual.Join	QualTable/Join permanently		Permanently joins two tables.

Table 8.1 (cont.): Description of Scripts

Script name	Menu	Icon	Function
Qual.Load	Qual/Load		Computes the load (kg/yr).
Qual.Mergetheme	QualTools/Merge themes		Merges two themes.
Qual.Pick	QualTools/Pick		Retrieves the values of grids for points in a point coverage.
Qual.Wshd	QualTools/Watershed		Delineates watersheds for points in a point coverage.
Qual.Zonalmean	QualTools/Zonalmean		Computes the averages within a zone.

8.1.3 Extension scripts

The extension scripts are system scripts: they do not appear when clicking on *Scripts* in the project window. However, the scripts can be obtained by loading the system scripts. To know the script corresponding to a given menu or button in a project (e.g. project austin.apr in Figure 8.4), the user has just to use the Customize window which appears when using the command *Project/Customize* accessible when the Project window is active. For example, in Figure 8.4, the script *Qual.Flow* is linked to the command *Qual/Flow* visible when a view in active (*Type: View*).

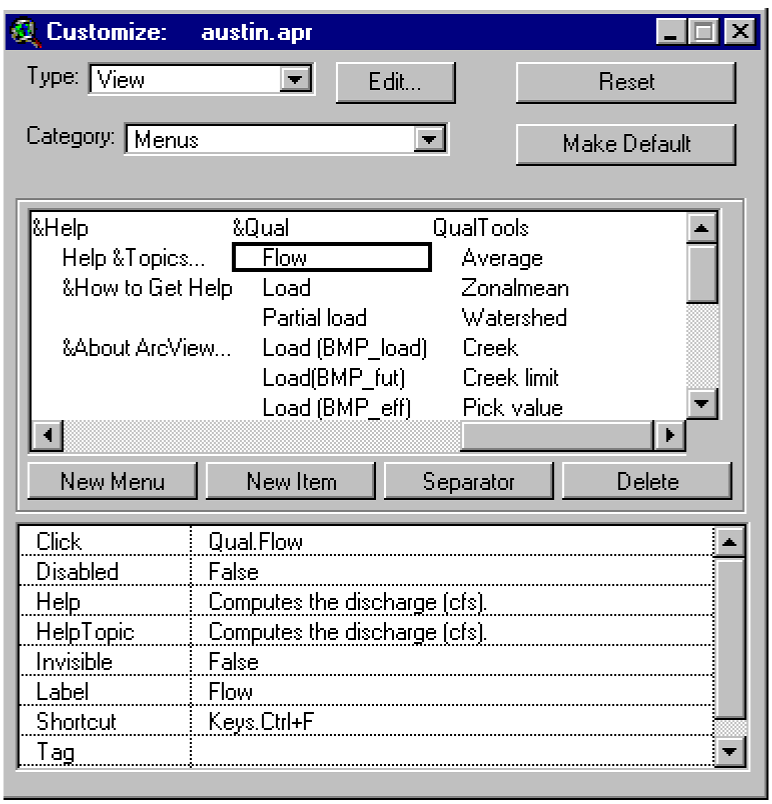


Figure 8.4: Script customization window

To view the script Qual.Flow, the user has then to add a new script by clicking first on *Scripts* and then on *New* in the project window. A Script window called *Script1* appears. The script *Qual.Flow* is obtained by using the command *Script/Load System Script* accessible when the Script window is active (Figure 8.5).

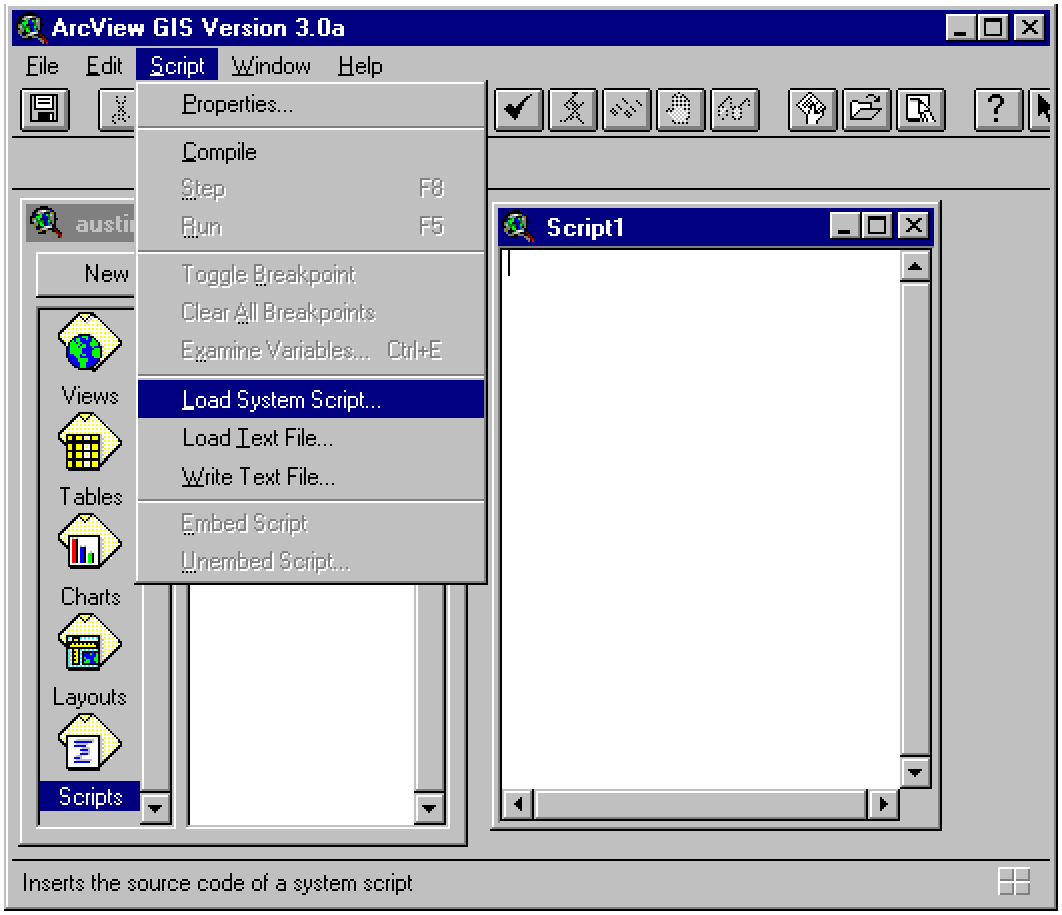


Figure 8.5: Load an extension script

The *Script Manager*, containing the list of system scripts in alphabetical order appears (Figure 8.6).

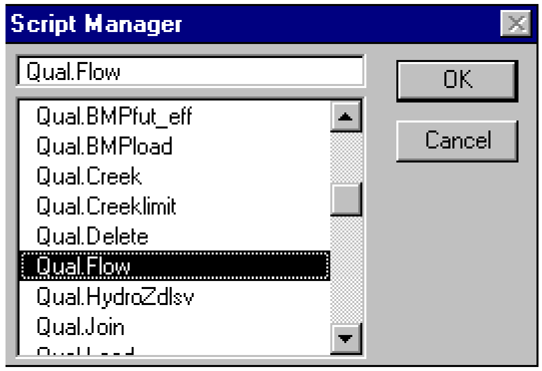


Figure 8.6: Script Manager

After selecting the desired script in the Script Manager, this script is written in the active Script window (*Script1*, Figure 8.7). The script can be renamed by using the command *Script/Properties*.

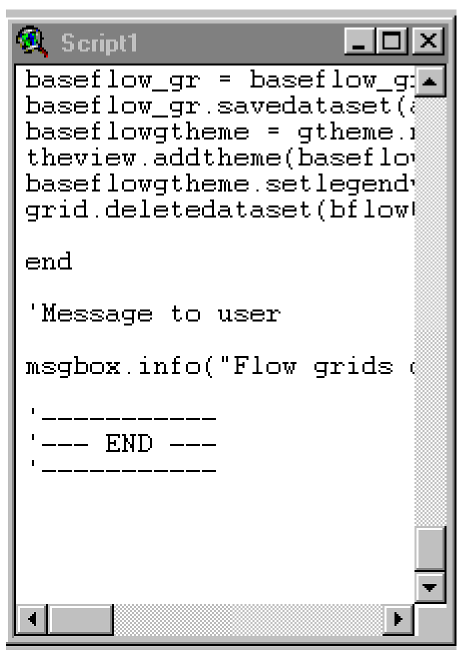


Figure 8.7: An Extension Script

8.1.4 Extension creation

An extension is created in two steps:

- Create a project with all the scripts and customization which will be in the extension.
- Write the scripts creating the extension.

- **Extension project**

First a project with all the components wanted in the extension is created. The project must look like the extension: the only difference is that the scripts are visible in the project whereas they are system scripts and hence invisible in the extension.

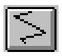
The first step is to add all the scripts needed to the project. It is easier when creating an extension to use script with the same prefix (e.g. Qual.). Once the scripts have been added, they need to be customized with buttons, menus and tools by using the *Customize* window (Figure 8.4).

- **Extension scripts**

Three scripts are needed to build an extension. One script is used to create the extension (Extension.Make): this script calls a script to install the extension (Extension.Install) and another to uninstall it (Extension.Uninstall). The Avenue scripts used to build the extension are in Appendix C. These scripts are easily corrected to modify the extension. The ArcView Online Help is also very helpful.

The remaining sections of this chapter show the user how to use the scripts for computing flows, loads and BMP effects.

8.2 FLOW COMPUTATION

Predicted flow, direct runoff and base flow are computed by using the script *Qual.Flow* (Appendix C). Note that this script can also be used to recalibrate the discharge. This script can be run when a view window is active either by clicking on the button  or using the command *Qual/Flow*.

8.2.1 Input/Output files

- **Qual.Flow inputs**

Before running the script, the input files needed in the script must be added in the View window. The number of input files depends whether a recharge zone is considered. If there is a recharge zone, five files are necessary; otherwise only three files are required (Figure 8.8).

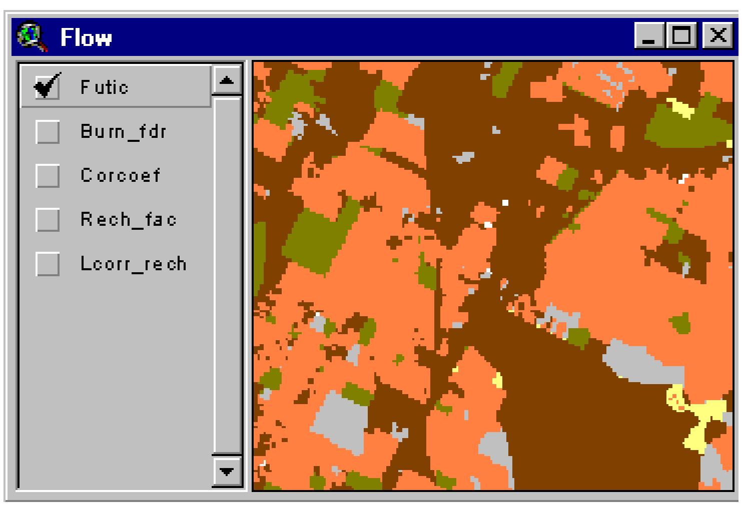


Figure 8.8: Input files (Qual.Flow)

1. Impervious cover coverage or grid (*futic,* Chapter 4). The attribute table of the coverage must contain an impervious cover field. The impervious cover can be expressed either as percentage or as decimal fraction.
2. Flow direction grid (*burn_fdr*, Chapter 3).
3. Flow correction coefficients grid (*corcoef*, Chapter 5).

The last two inputs are needed only if a recharge zone is taken into account.

4. Recharge flow grid in cfs (*rech_fac*, Chapter 5).
5. Cell recharge grid in cfs (lcorr_rech, Chapter 5).

Note that two scripts creating the grids needed for calibration (*corcoef* for flow calibration, *rech_fac* and *lcorr_rech* for recharge) are currently developed at CRWR and should be available by December 15, 1997.

- **Qual.Flow outputs**

The script creates six output grids if a recharge zone is considered (Figure 8.9) and five otherwise. These grids are:

1. Corrected direct runoff generated in each cell in cfs (*runcel1).*
2. Corrected base flow generated in each cell in cfs (*bflowc1).*
3. Total flow without considering recharge in cfs (*tflow01).* *
4. Total predicted flow in cfs (*flow1).*
5. Direct runoff in cfs (*runoff1).*
6. Base flow in cfs (*basefl1).*

* Grid computed only if a recharge zone is considered.

The names in italic are the names given by default to the outputs. The number at the end (1 in this case) enables one to differentiate the files created in different runs. For example if the file *runcel1* already exists, the new corrected direct runoff grid is called *runcel2*. All the grids created are added as themes in the view at the top of the legend. The theme at the top is hence the last file created.

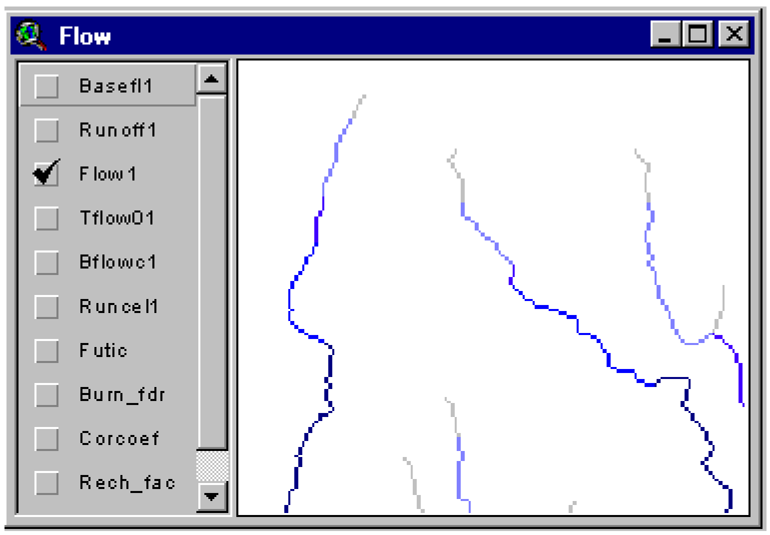
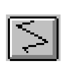


Figure 8.9: Output files (Qual.Flow)

8.2.2 Running Qual.Flow

Once the input files have been added to the View, the script can be run by clicking on the button  or by using the command *Qual.Flow*. The following section describes the series of message boxes appearing on the screen during the flow computation.

The first window appearing on the screen is the *Analysis Properties* window, which is normally accessed through the command *Analysis/Properties*. This window was inserted in the program so that the user has to define the analysis extent and the cell size before the computations begin. It is important since the analysis extent is set by default to the maximum grid extent, for which the computation length is quite long. Several options are available to choose an analysis extent (Figure 8.10). Note that option to set the extent to what can be seen in the *View* window is *Same as Display* and not *Same as View*.

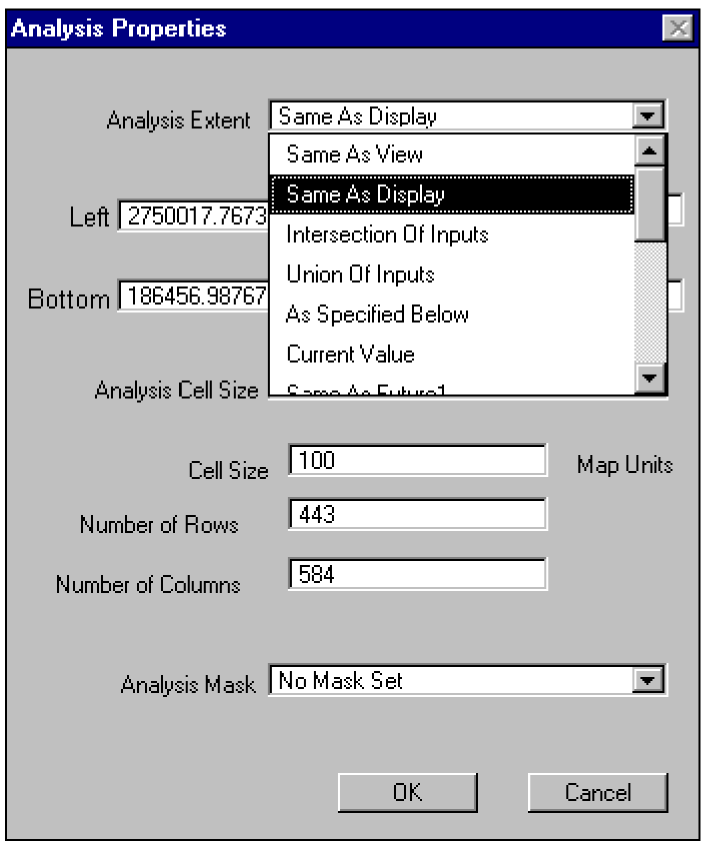


Figure 8.10: Analysis extent and cell size (Qual.Flow)

The following message box (Figure 8.11) prompts the user for the working directory where the temporary files and the output grids *runcel1*, *bflowc1* and *tflow01* will be written. Since these grids are used in the load and BMP computation. It is important to be able to locate easily these three files.

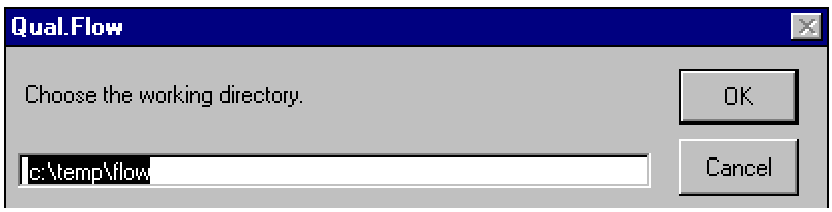


Figure 8.11: Working directory (Qual.Flow)

A series of message boxes prompts then for the input files. The first box prompts the user for the impervious cover theme (Figure 8.12).

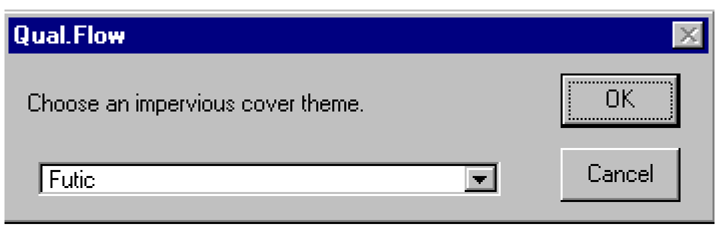


Figure 8.12: Choose an impervious cover theme (Qual.Flow)

If the theme chosen is a polygon coverage, then a message box prompts the user for the name of the impervious cover field in the attribute table (Figure 8.13). If it is a grid theme, the script automatically assumes that the impervious cover is the value given in the grid.

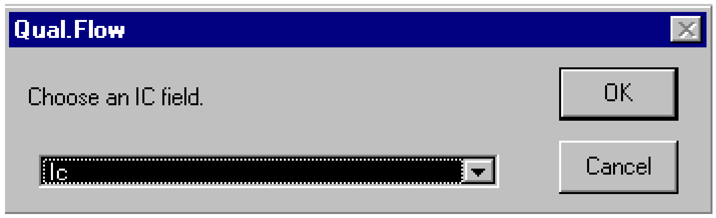


Figure 8.13: Choose an impervious cover field (Qual.Flow)

The second input file is a flowdirection grid (*burn_fdr*, Figure 8.14) which is used to compute flowaccumulation grids. The cell value indicates the direction of the flow. The grid is based on the topography of the study area (Chapter 3).

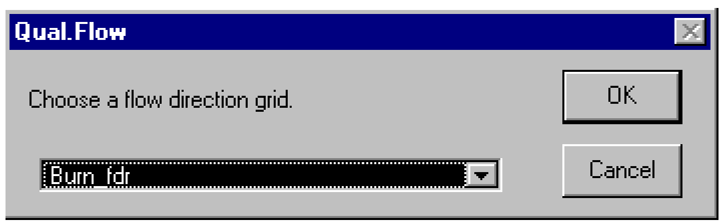


Figure 8.14: Choose a flowdirection grid (Qual.Flow)

The following input (Figure 8.15) is a flow correction grid (*corcoef*) obtained by flow calibration. This grid is based on the comparison between observed and predicted flows based on impervious cover/runoff coefficient relationships. If the volume of precipitation, the relationships used or the observed values used are modified for current conditions, the correction grid must be redefined (section 5.3.3).

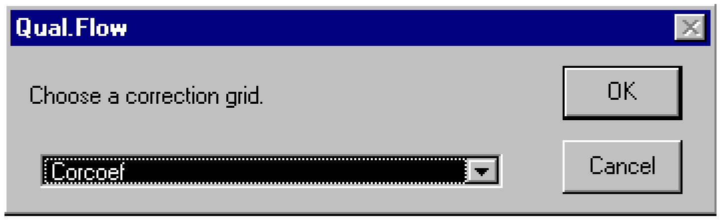


Figure 8.15: Choose a correction grid (Qual.Flow)

The user is then asked whether he or she wants to consider a recharge zone (Figure 8.16). The recharge zone makes the computation longer and more complicated.

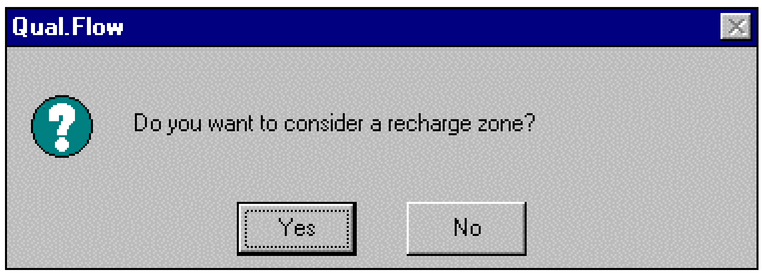


Figure 8.16: Recharge zone (Qual.Flow)

When considering a recharge zone, two more inputs are needed to model the recharge flow. The first (Figure 8.17) is a grid with the recharge lost in each cell within the creeks located in the recharge zone (*lcorr_rech*, section 5.1.4). The second input (*rech_fac*, Figure 8.8) is the result of a flow accumulation on the grid. This grid indicates the total recharge flow lost upstream of a given cell.

Figure 8.17: Choose a cell recharge grid (Qual.Flow)

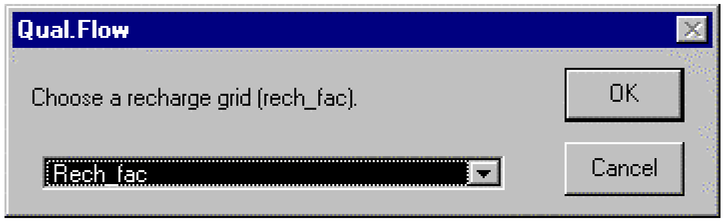


Figure 8.18: Choose a recharge grid (Qual.Flow)

The user is then asked for the names to give to the direct runoff (*runoff1*, Figure 8.19), base flow (*bflow1*) and total flow (*flow1*) grids.

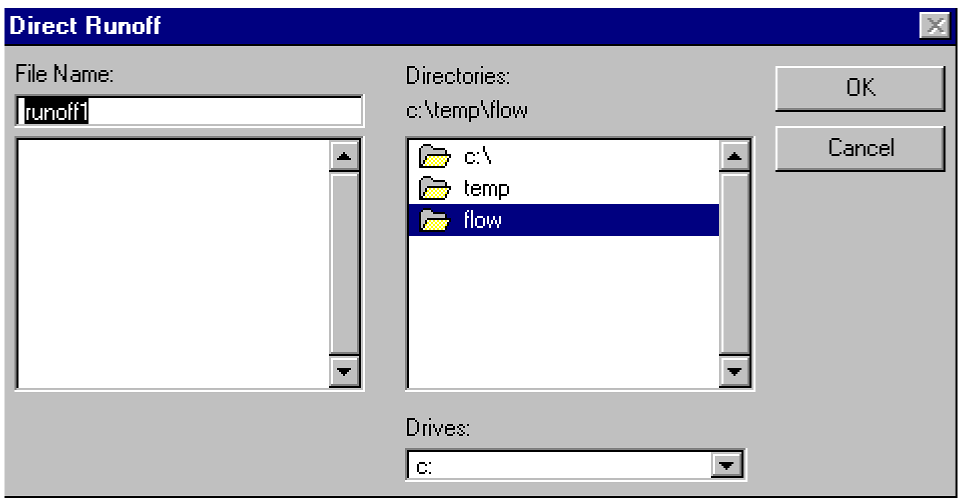


Figure 8.19: Name of the output grid (Qual.Flow)

If the impervious cover theme is a polygon theme, the runoff coefficients may have already been calculated during a previous run. It is perhaps not necessary to recompute them. A message box asks the user whether he wants to redo the computation (Figure 8.20).

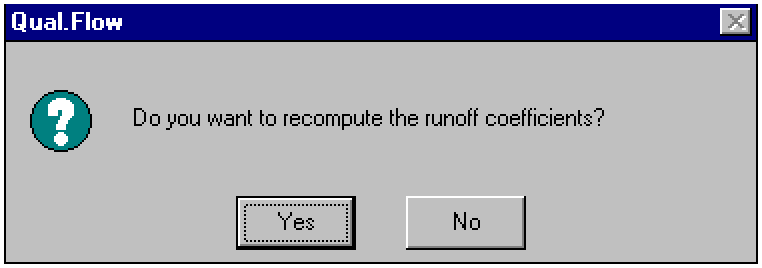


Figure 8.20: Recompute the runoff coefficients (Qual.Flow)

By default, the names given to the runoff coefficient fields are *runcoef* for direct runoff and *runcoef_bf* for base flow. If a field already exists, a message box asks the user if he wants to overwrite this field (Figure 8.21).

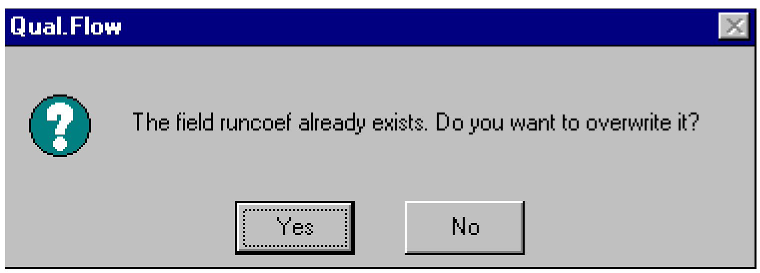


Figure 8.21: Overwrite an existing field (Qual.Flow)

By answering "no", a message box appears which prompts the user for the name to give to the new field (Figure 8.22). Note that the runoff coefficients fields used if the coefficients are not recomputed are the fields *runcoef* and *runcoef_bf*.

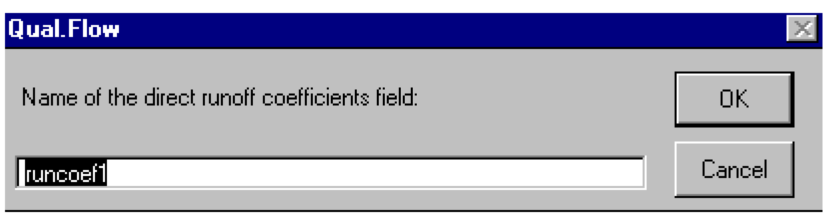


Figure 8.22: Renaming a runoff coefficient field (Qual.Flow)

If the runoff coefficients are computed, a series of message boxes prompt the user for the impervious cover/runoff coefficient relationships to use. First, a message box shows the default relationship for direct runoff (Figure 8.23) presented in Chapter 5. The message box also gives the user the possibility to modify the relationship without modifying the script.

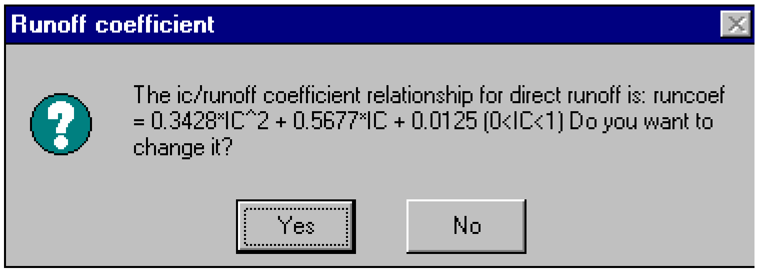


Figure 8.23: Default impervious cover/direct runoff coefficient relationship (Qual.Flow)

The relationship is defined as a second order equation whose coefficients can be modified by the user (Figure 8.24). The impervious cover values in the equation are given as decimal fractions. If the format of the equation changes (third order…) the script must be modified to accommodate that.

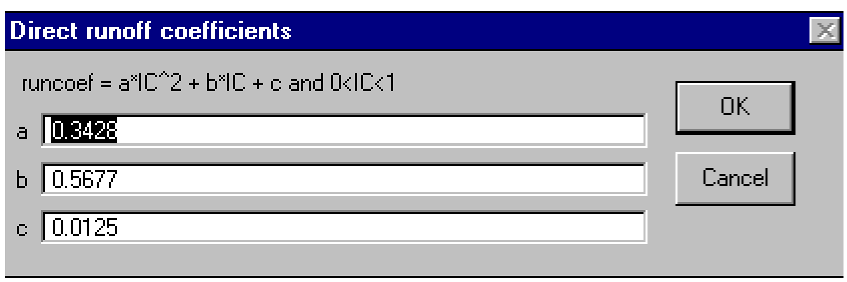


Figure 8.24: Modifying the impervious cover/direct runoff coefficient relationship (Qual.Flow)

The process is similar for base flow runoff coefficients. A message box shows the default relationship defined in the study (Figure 8.25).

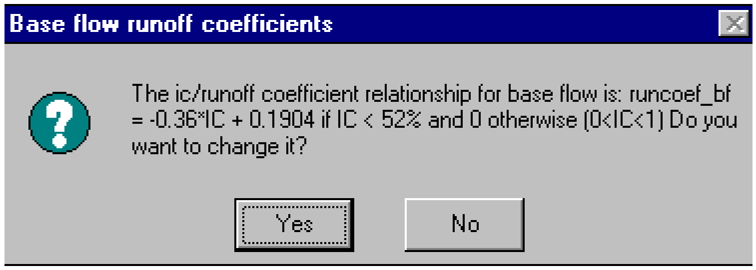


Figure 8.25: Default impervious cover/base flow runoff coefficients relationship (Qual.Flow)

The relationship is defined as a first order equation whose coefficients can be modified (Figure 8.26). The impervious cover value above which no base flow occurs is automatically modified. However, if the format of the equation changes (second order…), the script must be modified and the Extension must be recreated.

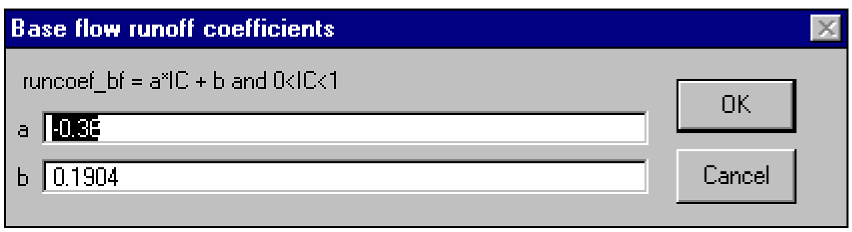
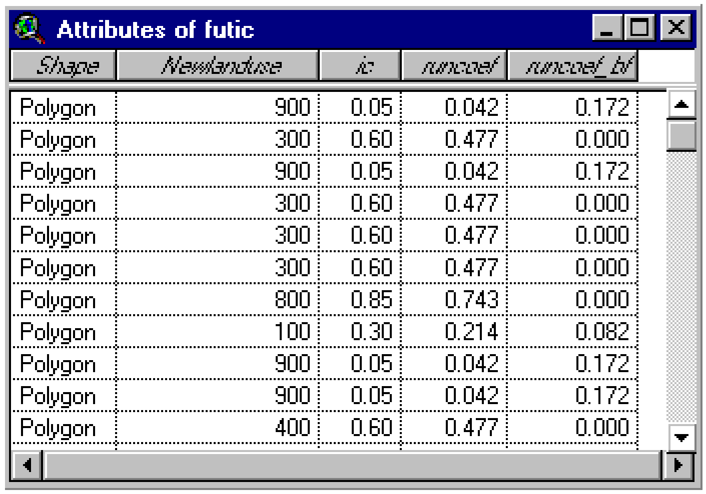


Figure 8.26: Modifying the impervious cover/base flow coefficients relationship (Qual.Flow)

If the impervious cover is a polygon coverage, its attribute table is edited to add the runoff coefficients fields (Figure 8.27).



Attributes of futic

Shape	Newlanduse	ic	runcoef	runcoef_bf
Polygon	900	0.05	0.042	0.172
Polygon	300	0.60	0.477	0.000
Polygon	900	0.05	0.042	0.172
Polygon	300	0.60	0.477	0.000
Polygon	300	0.60	0.477	0.000
Polygon	300	0.60	0.477	0.000
Polygon	800	0.85	0.743	0.000
Polygon	100	0.30	0.214	0.082
Polygon	900	0.05	0.042	0.172
Polygon	900	0.05	0.042	0.172
Polygon	400	0.60	0.477	0.000

Figure 8.27: Impervious cover attribute table (Qual.Flow)

The script then calculates the new grids. A message box indicates that the computation has been completed.

For an undetermined reason, the error message box shown in Figure 8.28 appears sometimes. The only solution to eliminate this problem is to save the project and exit from ArcView. The script should then run without any problem. However, if it still does not work, the solution is to exit from ArcView again.

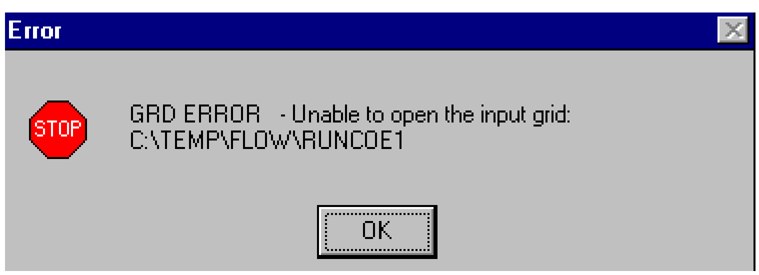


Figure 8.28: Error message (Qual.Flow)

8.3 LOAD COMPUTATION

The predicted external load for any constituent for which the event mean concentrations are known is computed with the script *Qual.Load.* This script can be run when a View window is active either by clicking on the button or using the command *Qual/Load*.

8.3.1 Input/Output files

- **Qual.Load inputs**

Two tables (Figure 8.31) and seven themes (Figure 8.29) are needed as inputs to the script if a recharge zone is to be considered.

1. Direct runoff EMC table (*emcrun.dbf*, section 8.3.1)
2. Base flow EMC table (*emcbf.dbf*, section 8.3.1)
3. Impervious cover coverage or grid (*futic*, Chapter 4). The attribute table of the coverage must contain an impervious cover field. The impervious cover can be expressed either as percentage or as decimal fraction. The program examines the maximum impervious cover value: if this value is less than the impervious cover is considered as decimal fraction.
4. Flow direction grid (*burn_fdr*, Chapter 3)
5. Water land use zones grid (*zone_gr*, Chapter 6). Water must be given the grid-code 999 (water code for traffic serial zones).
6. Corrected direct runoff generated in each cell in cfs (*runcel1*). This grid is an output of the flow computation.

7. Corrected base flow generated in each cell (*bflowc1*). This grid is an output of the flow computation.

The next two grids are required only if a recharge zone is considered.

8. Total flow without considering recharge in cfs (*Tflow01*).

9. Cell recharge in cfs (*lcorr_rech*, Chapter 5).

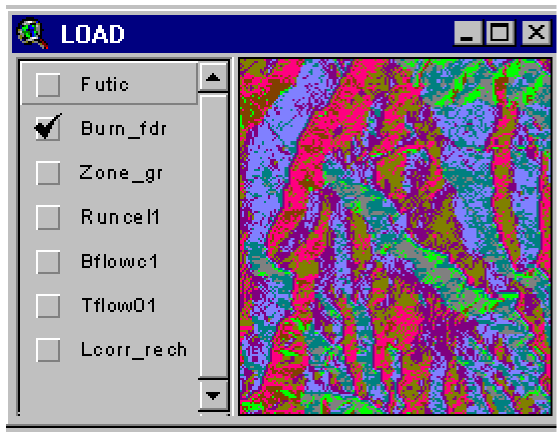


Figure 8.29: Input themes (Qual.Load)

- **Qual.Load outputs**

The script creates as many load grids as number of constituents for which the load is computed. The constituents are chosen by the user in a list based on the constituents in the EMC tables. Figure 8.30 shows the view obtained after running the script for BOD, COD and Cu.

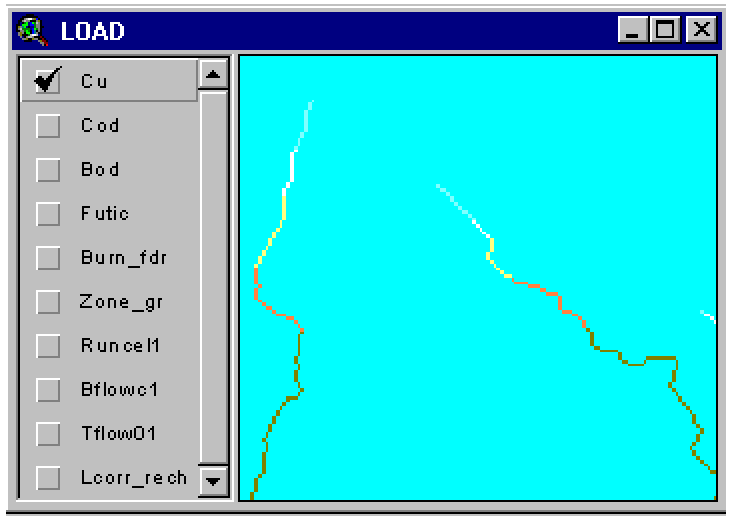


Figure 8.30: Inputs and outputs (Qual.Load)

- **Format for EMC tables**

Tables containing the coefficients of the impervious cover/EMCs relationships are needed to compute the load. The tables must be added to the project by clicking first on the Project window (*qual.apr*, Figure 8.31), then on *Tables* and finally on *Add*. Two tables are necessary: one for direct runoff EMCs (*emcrun.dbf*) and the other for base flow EMCs (*emcbf.dbf*).

Direct runoff

Linear relationships were developed between impervious cover and direct runoff EMCs (Chapter 6):

EMC = a+ b*IC, where 0<IC<1 and EMC in mg/l

Two coefficients, a and b, are needed for each constituent to define the relationship. The direct runoff EMCs table *emcrun.dbf* contains three fields (Figure

8.32). The first field contains the names of the constituents, the second the value of the constant a and the third the value of the first order coefficient b.

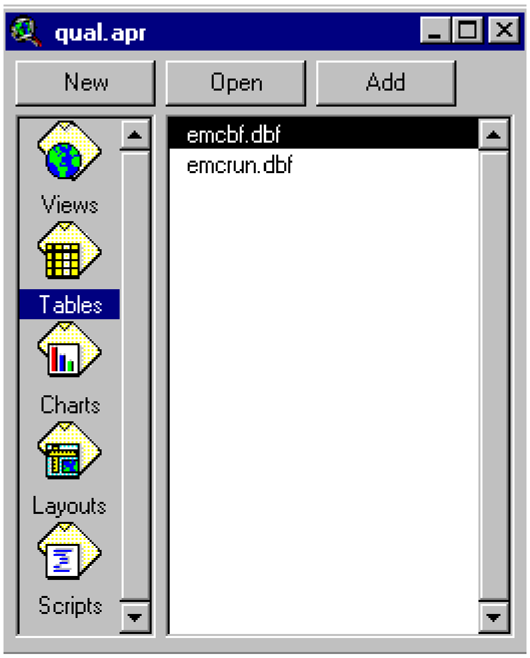
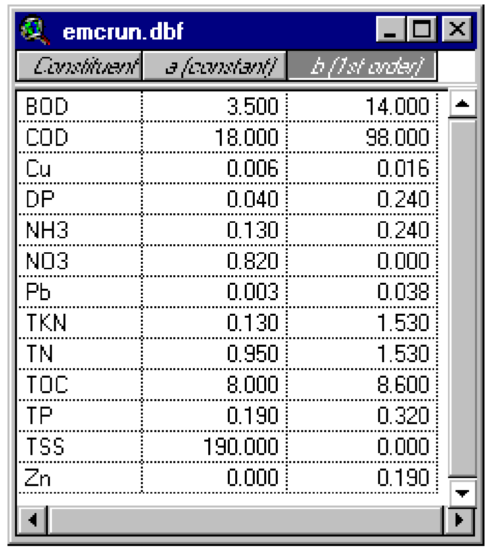


Figure 8.31: Add tables (Qual.Load)


emcrun.dbf

Constituent	a (constant)	b (1st order)
BOD	3.500	14.000
COD	18.000	98.000
Cu	0.006	0.016
DP	0.040	0.240
NH3	0.130	0.240
NO3	0.820	0.000
Pb	0.003	0.038
TKN	0.130	1.530
TN	0.950	1.530
TOC	8.000	8.600
TP	0.190	0.320
TSS	190.000	0.000
Zn	0.000	0.190

Figure 8.32: Direct runoff EMCs table (Qual.Load)

Polynomial relationships higher than first order can be used without modifying the script. The values of the n^{th} order coefficients must be written in the $(n+2)^{th}$ column. However, for any other relationship, the script and the table needs to be modified.

Base flow

The format of this table is a little different since the base flow EMCs are defined as constant values for undeveloped and developed areas (Chapter 6). Areas with an impervious cover less than 15% are considered undeveloped. This limit can be modified by setting the variable *undev* to a different value (in percentage) in the script Qual.Load.

$$EMC_{bf} = \begin{cases} c \text{ if } IC \leq undev \\ d \text{ if } IC > undev \end{cases}$$

The first column in the base flow EMC table contains the names of the constituents studied. This table must be consistent with the direct runoff EMC tables. Both tables must have the same number of constituents having exactly the same name and be in the same order. Error messages appear if the tables are not consistent. The second column contains the base flow EMCs in mg/l for undeveloped areas and the third one the base flow EMCs in mg/l for developed areas (Figure 8.33).

emcbf.dbf

Constituent	Undeveloped	Developed
BOD	0.450	0.800
COD	12.000	20.000
Cu	0.000	0.000
DP	0.014	0.060
NH3	0.020	0.060
NO3	0.150	0.600
Pb	0.000	0.000
TKN	0.280	0.460
TN	0.430	1.060
TOC	2.000	5.000
TP	0.020	0.120
TSS	0.000	0.000
Zn	0.000	0.000

Figure 8.33: Base flow EMC table (Qual.Load)

8.3.2 Running Qual.Load

Once the input tables have been added to the Project window and the input themes to the view, the script *Qual.Load* can be run by clicking on the button  or by using the command *Qual.Flow*.

As for the flow computation, the first window which appears is the *Analysis Properties* window prompting the user for the analysis extent and for the cell size to use. To keep the same extent and cell size used in the flow computation, if they have not been modified, they just need to be set to *current value* in the *Analysis Properties* window.

Also as for the flow computation, the following message box prompts the user for the name of the working directory where the temporary files will be written.

Next the script prompts the user for the EMC tables, first for direct runoff (*emcrun.dbf*, Figure 8.34) and then for base flow (Figure 8.35).

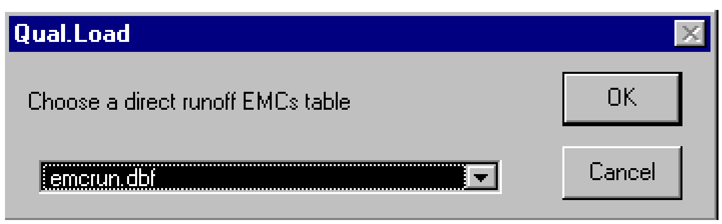


Figure 8.34: Choose a direct runoff EMCs table (Qual.Load)

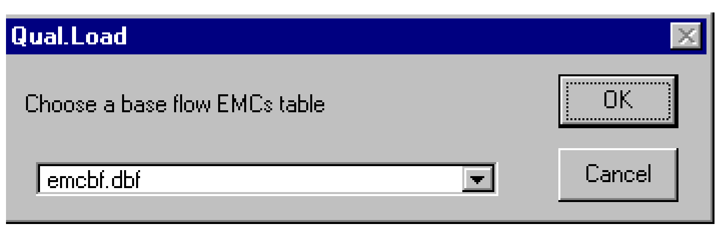


Figure 8.35: Choose a base flow EMCs table (Qual.Load)

After selecting the EMCs tables, the script prompts the user for the constituents to model (*emcbf.dbf*, Figure 8.36). Several constituents may be selected.

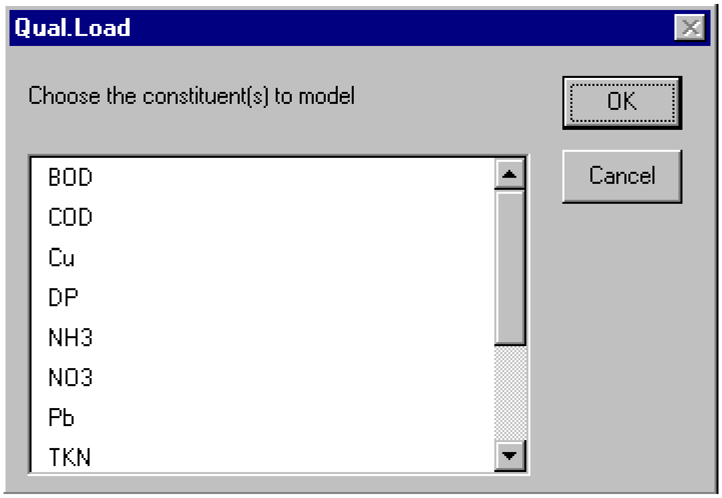


Figure 8.36: Choose the constituent(s) to model (Qual.Load)

For each selected constituent a message box prompts the user for the name to give to the constituent load grid (e.g. BOD in Figure 8.37).

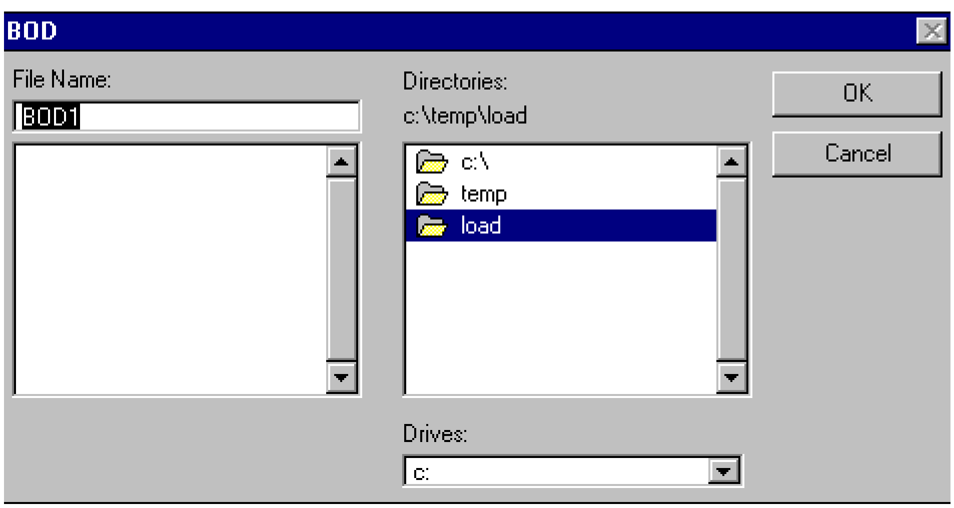


Figure 8.37: Name the load grid (Qual.Load)

A series of message boxes prompts then for the input themes. The first input theme is the impervious cover theme (*futic*, Figure 8.38). If the impervious cover is a polygon coverage then the user is prompted for the name of the impervious cover field.

Figure 8.38: Choose an impervious cover theme (Qual.Load)

The user is then prompted for a flowdirection grid (*burn_fdr*, Figure 8.39).

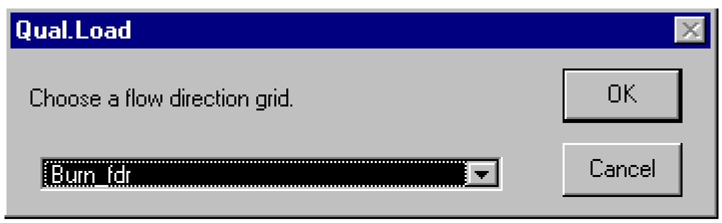


Figure 8.39: Choose a flowdirection grid (Qual.Load)

The following message box prompts for a water land use zones grid (*zone*_gr, Figure 8.40). This theme is used to correct the EMC grids so that the EMC corresponding to water is zero. For current conditions, water is defined by the land use code 940. With the traffic serial zones for future conditions, water is defined by the traffic serial zone number 999.

Figure 8.40: Choose a water land use theme (Qual.Load)

The script prompts then for a corrected cell runoff grid (*runcel1*, Figure 8.41). This grid is an output of the flow computation. It represents the direct runoff generated in any cell in cfs after applying the flow correction (but not the recharge correction if applicable).

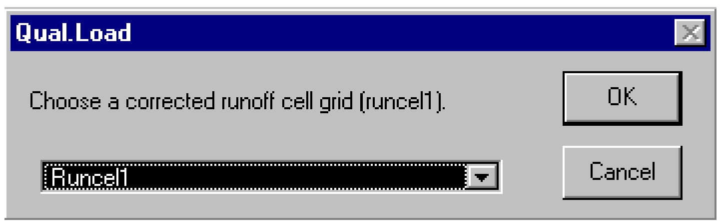


Figure 8.41: Choose a corrected runoff cell grid (Qual.Load)

The following message box prompts the user for a corrected base flow runoff grid (Figure 8.42). This grid is also an output of the flow computation. It corresponds to the amount of base flow in cfs generated by any cell after applying the flow correction (but not the recharge correction if applicable).

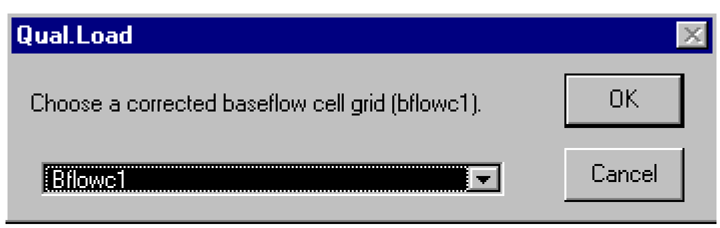


Figure 8.42: Choose a corrected base flow cell grid

A message box asks then the user whether he wants to consider a recharge zone (Figure 8.43).

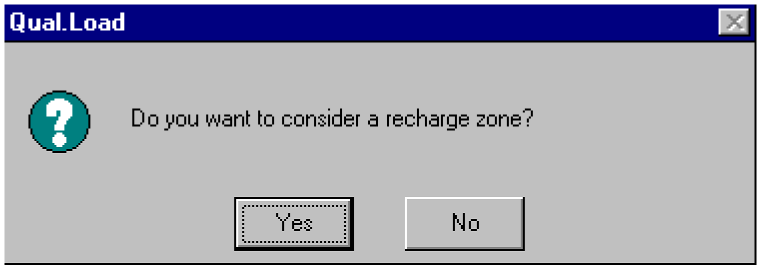


Figure 8.43: Recharge zone (Qual.Load)

If the answer is “no” then no other inputs are needed. Otherwise two more input themes are needed. The first input needed is a cell recharge grid which indicates the recharge lost in any cell (Figure 8.44). This grid is used to compute the load lost in the recharge (multiplied by the corresponding concentrations).

The second theme needed is a total flow grid, without considering the recharge (Figure 8.45). This grid is computed during the flow computation. It is used to compute the concentration in the flow, assuming that the concentration is the same with and without a recharge zone.

Figure 8.44: Choose a cell recharge grid (Qual.Load)

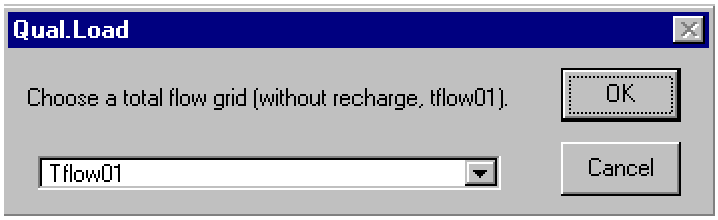


Figure 8.45: Choose a grid with the flow without recharge (Qual.Load)

The script computes then the load for each constituent selected and adds the new load grids to the view. A final message indicates that the load computation is finished.

8.4 BEST MANAGEMENT PRACTICES

Best Management practices can be modeled in three different ways:

- Located BMPs defined by mass removed (section 8.4.1).
- Located BMPs defined by efficiency (section 8.4.2).
- Non located BMPs defined by efficiency (section 8.4.3).

8.4.1 Located BMPs defined by load removed

The script *Qual.BMPload* (Appendix C) allows the user to compute the load removed by a set of BMPs described in a point coverage. It can be run either with the button  or the command *Qual/BMPload*. Two input files are needed to run the script (Figure 8.46):

- a flowdirection grid (*burn_fdr*, Chapter 3)
- a BMP point coverage (*city1*, Chapter 7). The attribute table must contain a table with a load removed field. The value in this field can be negative (e.g. NO_3).

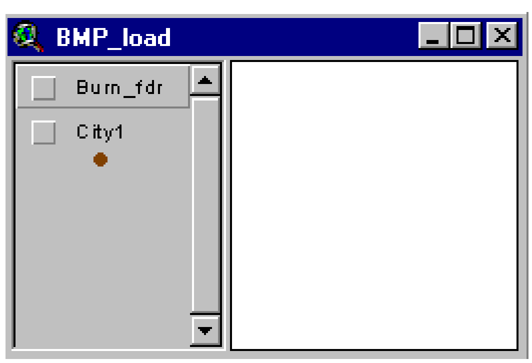


Figure 8.46: Input files (Qual.BMPload)

When running the script, a message box prompts first the user for the name of the working directory (Figure 8.47).

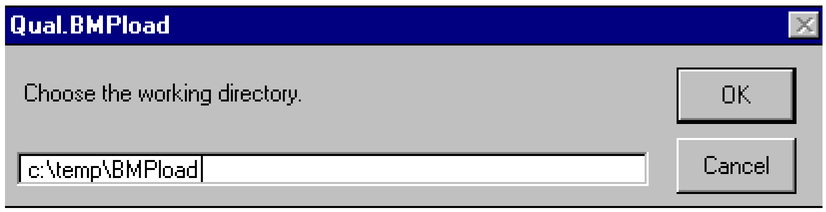


Figure 8.47: Choose the working directory (Qual.BMPload)

The following message box prompts the user for the name to give to the grid representing the load removed (Figure 8.48).

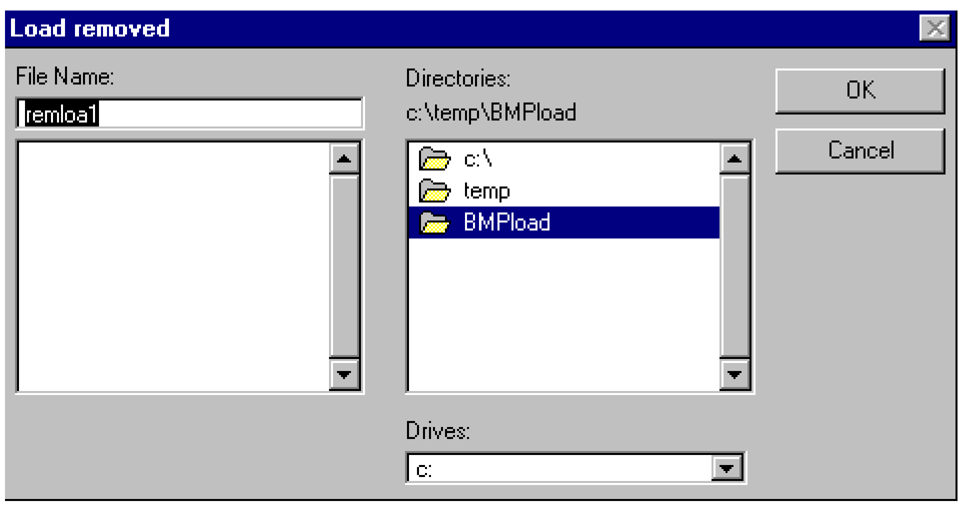


Figure 8.48: Name of the removed load grid (Qual.BMPload)

Two message boxes prompt then for a flowdirection grid (Figure 8.49) and a BMP point coverage (Figure 8.50).

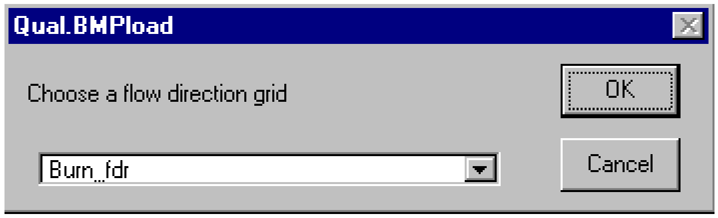


Figure 8.49: Choose a flowdirection grid

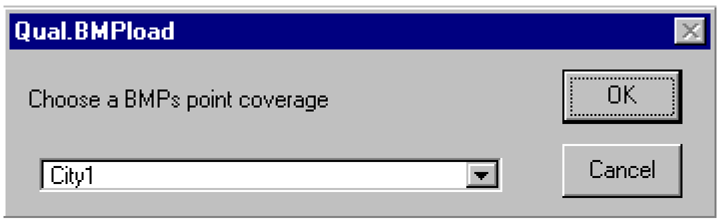
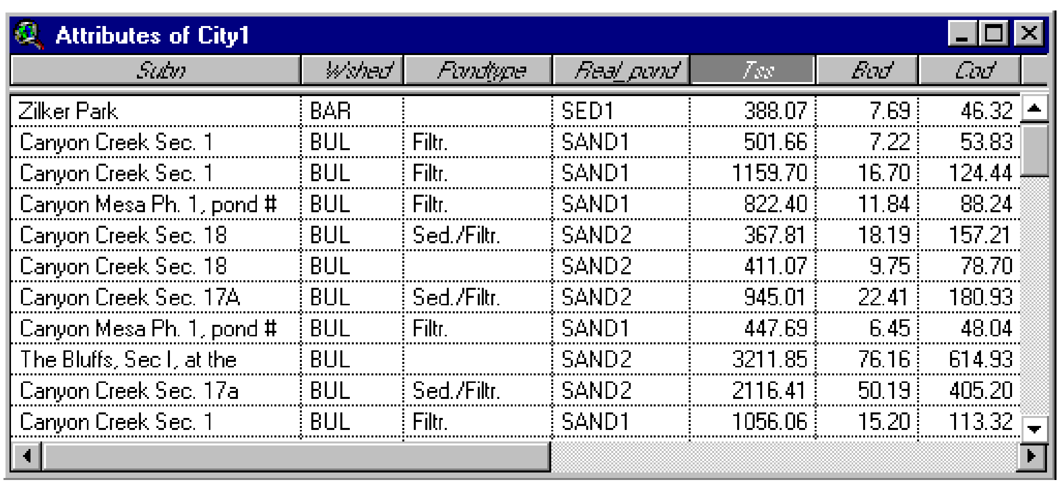


Figure 8.50: Choose a BMP point coverage (Qual.BMPload)

The next step consists of choosing in the attribute table of the point coverage the field containing the values of the loads removed. Figure 8.51 shows the attribute table of the point coverage *city1*which contains 121 residential BMPs.


Attributes of City1

Subn	Wshed	Pondtype	Real_pond	Tss	Bod	Cod
Zilker Park	BAR		SED1	388.07	7.69	46.32
Canyon Creek Sec. 1	BUL	Filtr.	SAND1	501.66	7.22	53.83
Canyon Creek Sec. 1	BUL	Filtr.	SAND1	1159.70	16.70	124.44
Canyon Mesa Ph. 1, pond #	BUL	Filtr.	SAND1	822.40	11.84	88.24
Canyon Creek Sec. 18	BUL	Sed./Filtr.	SAND2	367.81	18.19	157.21
Canyon Creek Sec. 18	BUL		SAND2	411.07	9.75	78.70
Canyon Creek Sec. 17A	BUL	Sed./Filtr.	SAND2	945.01	22.41	180.93
Canyon Mesa Ph. 1, pond #	BUL	Filtr.	SAND1	447.69	6.45	48.04
The Bluffs, Sec I, at the	BUL		SAND2	3211.85	76.16	614.93
Canyon Creek Sec. 17a	BUL	Sed./Filtr.	SAND2	2116.41	50.19	405.20
Canyon Creek Sec. 1	BUL	Filtr.	SAND1	1056.06	15.20	113.32

Figure 8.51: BMPs point coverage attributes table

For example the field TSS, which gives the loads of TSS removed at the BMP stations in the point coverage *city1* (residential BMPs), can be selected (Figure 8.52). The script can handle both positive and negative value.

Figure 8.52: Choose a removed load field (Qual.BMPload)

The script computes then the total load removed by the BMPs by doing a weighted flowaccumulation (Chapter 7).

The message box “Input grid has errors” appears for an obscure reason from time to time. The solution is to exit from ArcView and to reopen the project until the script works.

A message box indicates that the removed load grid has been computed (Figure 8.53) and added to the View (Figure 8.54).

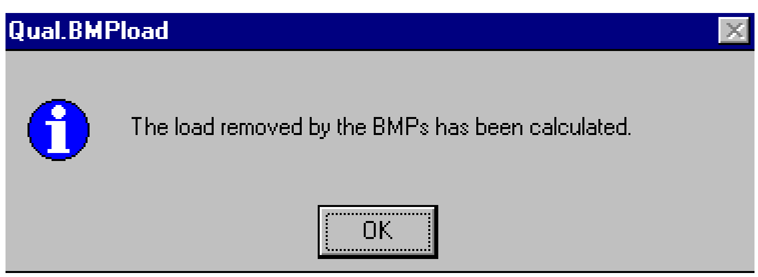


Figure 8.53: Final Message to user (Qual.BMPload)

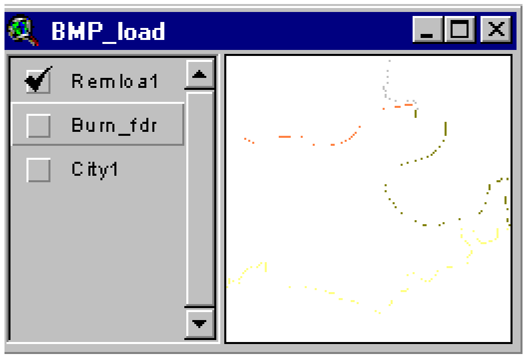


Figure 8.54: Input and output themes (Qual.BMPload)

8.4.2 Located BMPs defined by efficiency

The script *Qual.BMPeff* allows the user to compute the load removed by a set of BMPs defined by efficiency. The script can be run by clicking on the button  or by using the command *Qual/BMPeff*. This script is based on the principle described in Chapter 7. Like the previous script, *Qual.BMPeff* computes the load removed by located BMPs defined in a point coverage. The difference is that the BMPs are defined by efficiency instead of load removed. Hence the load removed at any BMP depends on what was removed upstream of that BMP. The load removed is computed as the difference between the original load and the removed load.

Note that the script can not handle negative efficiencies yet and needs to be modified to do so.

- **Qual.BMPeff input files**

Five input files are needed (Figure 8.55):

1. Flowdirection grid (*burn_fdr*, Chapter 3).
2. Flowaccumulation grid (*burn_fac*, Chapter 3).
3. BMPs point coverage (*BMP.shp*, Chapter 7). The attribute table must contain an efficiency field. The efficiencies can be expressed either as percentage or decimal fraction. If the efficiencies must be corrected, the table must also contain a field with the observed drainage areas in acres.
4. Initial load grid (e.g. *BOD).* This grid is the result of the load computation.

5. Watershed zones grid (*wshdzone*, *Chapter 7*). This grid is used to determine if two watersheds are nested. If two BMPs are located in two non-nested watersheds, then these two BMPS are totally independent (section 7.1.2) and can be handled at the same time.

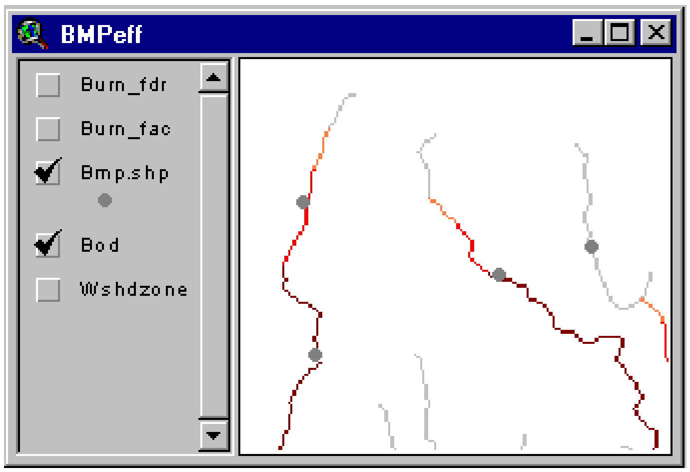
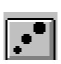


Figure 8.55: Input files (Qual.BMPeff)

- **Running Qual.BMPeff**

After adding the input files to the view, the script can be run by clicking on the button  or by using the command *Qual/BMPeff*.

The user is first asked to set the analysis extent and the cell size in the *Analysis Properties* window. The following message box prompts the user for the name of the working directory where the temporary files will be written (Figure 8.56).

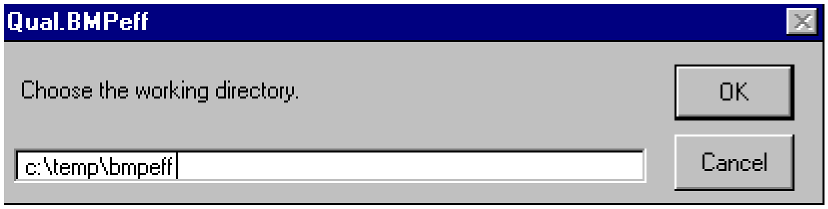


Figure 8.56: Choose the working directory (Qual.BMPeff)

The script prompts then for the name of the new load grid, which is the output of the computation (Figure 8.57).

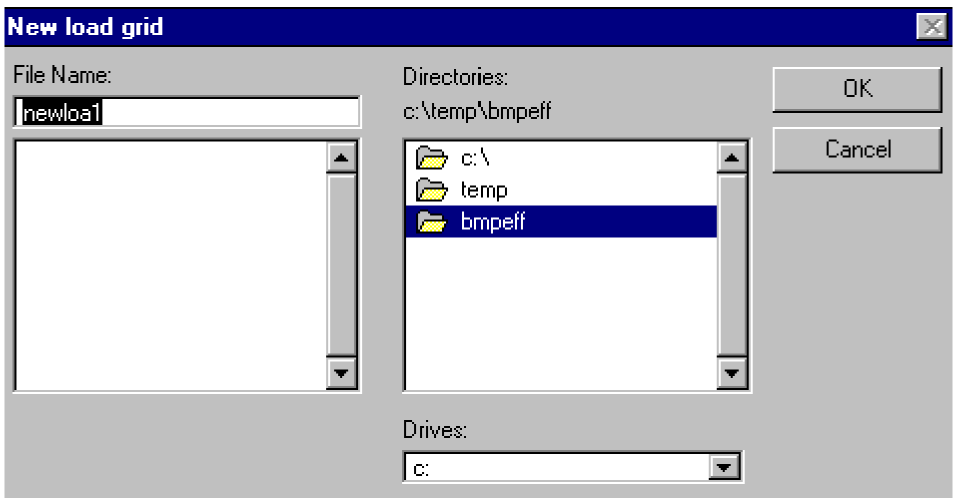


Figure 8.57: Name of the output grid (Qual.BMPeff)

A series of message boxes then prompt the user for the input files. The first input is a flowdirection grid (Figure 8.58).

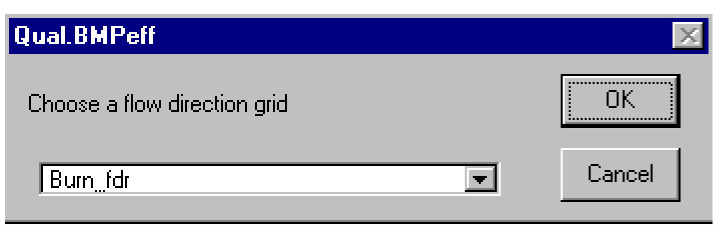


Figure 8.58: Choose a flow direction grid (Qual.BMPeff)

The second file theme needed is a flowaccumulation grid (Figure 8.59), used to determine the order in which the BMPs must be considered (Chapter 7). This grid is also used to obtain the drainage areas.

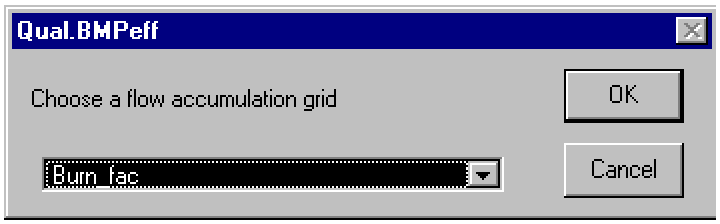


Figure 8.59: Choose a flowaccumulation grid (Qual.BMPeff)

The next input file is the BMP point coverage (Figure 8.60).

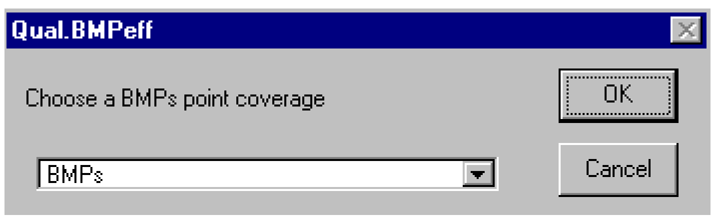


Figure 8.60: Choose a BMP point coverage (Qual.BMPeff)

The attribute table of the coverage must contain an efficiency field (Figure 8.61, *BOD_eff* for BOD).

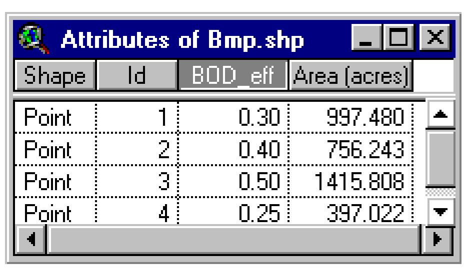


Figure 8.61: BMP point coverage attribute table (Qual.BMPeff)

The following message box (Figure 8.62) prompts the user for the name of the field containing the efficiency values in the point coverage attribute table. The script can handle efficiencies written as percentage or decimal fraction. Every value in a field must be defined however with the same format.

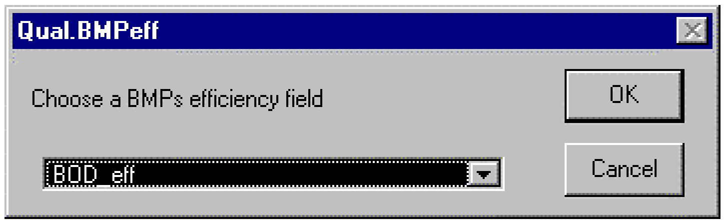


Figure 8.62: Choose a BMP efficiency field (Qual.BMPeff)

The user is then asked whether he or she wants to correct the efficiencies (Figure 8.63).

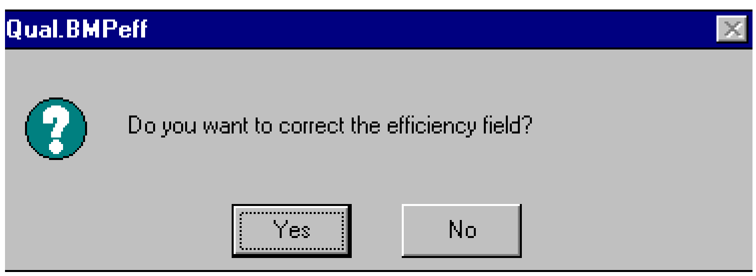


Figure 8.63: Correct the efficiency field (Qual.BMPeff)

The correction is based on a comparison between observed and modeled drainage areas (Chapter 7). If “yes” is the answer to the message box, the script prompts the user for the observed drainage areas (in acres) field (Figure 8.64).

Figure 8.64: Choose an observed drainage area (Qual.BMPeff)

To get the values of the initial load at the BMPs locations, an initial load grid is necessary (Figure 8.65).

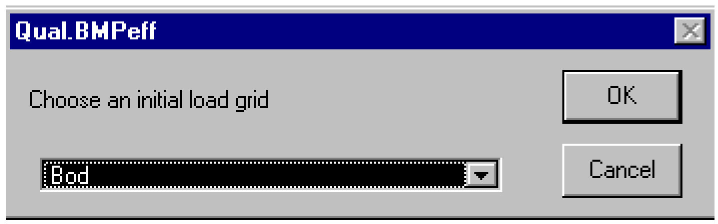


Figure 8.65: Choose an initial load grid (Qual.BMPeff)

Finally, the user is prompted for a watershed zones grid (*wshdzone*, Figure 8.66). This grid is used to define the nested watersheds (section 7.1.2).

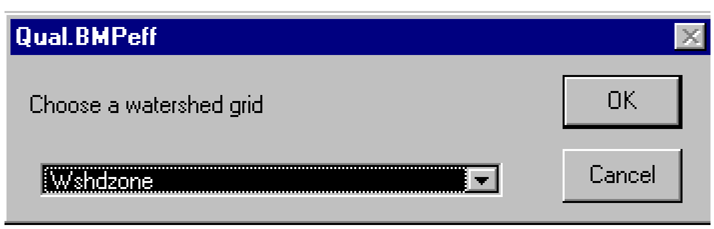


Figure 8.66: Choose a watershed zones grid (Qual.BMPeff)

The script computes then the new load after the effect of the BMPs. A message box announces the completion of the program (Figure 8.67) and the new grid is added to the view (Figure 8.68).

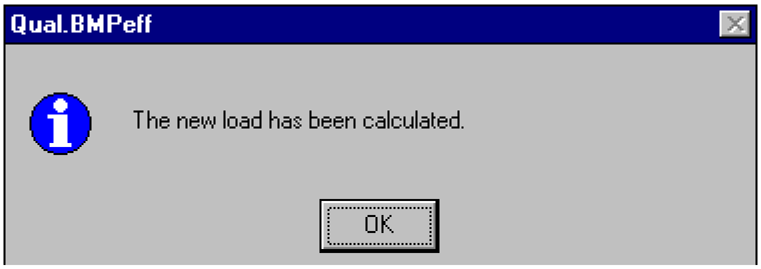


Figure 8.67: Final message to user (Qual.BMPeff)

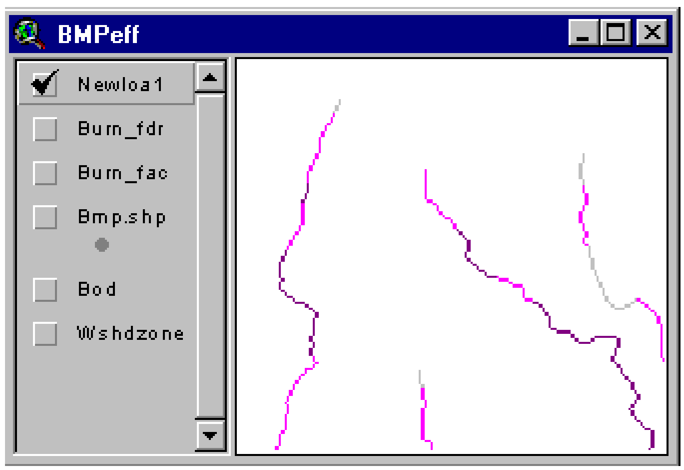
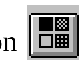


Figure 8.68: Input and output files (Qual.BMPeff)

8.4.3 Non located BMPs defined by efficiency

The script *Qual.BMPfut* allows the user to model the effect of non located BMPs defined by efficiency. This script is customized with the button  and the command *Qual/BMPfut*.

- **Qual.BMPfut input files**

Four tables (Figure 8.69) and nine themes plus a load grid for each constituent considered (e.g. for two constituents BOD and COD in Figure 8.70). The last three input themes (*Flow1*, *Tflow01* and *Lcorr_rech* in Figure 8.70) are needed only if a recharge zone is considered.

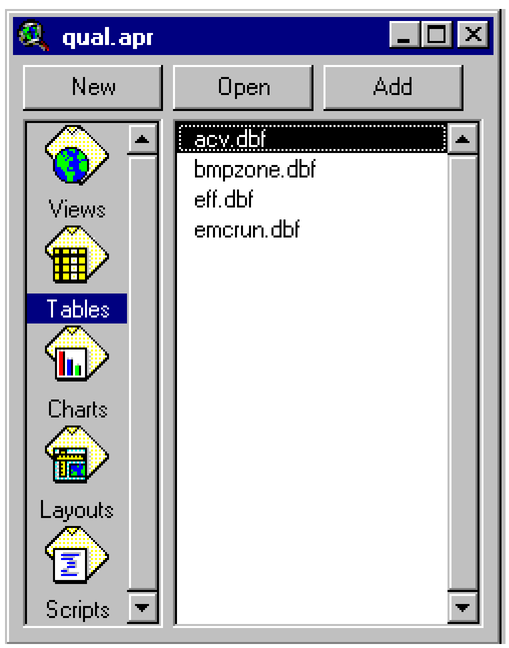


Figure 8.69: Input tables (Qual.BMPfut)

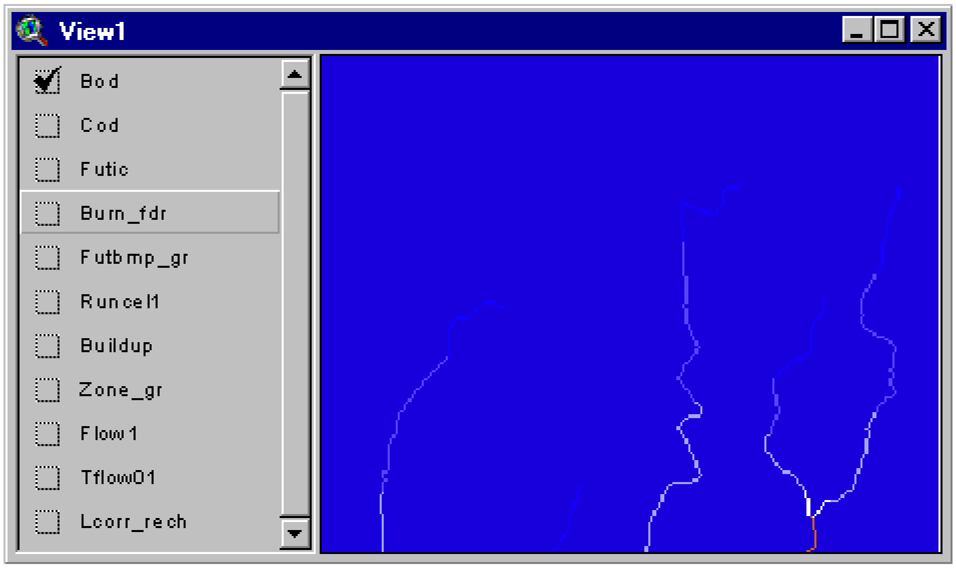


Figure 8.70: Inputs themes (Qual.BMPfut)

1. Average capture volume table (*acv.dbf*, section 8.4.3).

2. BMP zones table (*bmpzone.dbf*, section 8.4.3).

3. Efficiency table (*eff.dbf*, section 8.4.3).

4. Direct runoff EMC table (*emcrun.dbf*, section 8.3.1).

5. Initial load grid (e.g. *BOD*, *COD*) for each constituent considered. The load grids are computed with the script *Qual.Load*.

6. Impervious cover coverage or grid (*futic*, Chapter 4). The attribute table of the coverage must contain an impervious cover field. The impervious cover can be expressed either as percentage or as decimal fraction.

7. Flow direction grid (*burn_fdr*, Chapter 3).

8. BMP zones grid (*futbmp_gr*, Chapter 7).

9. Corrected direct runoff generated in each cell (*runcel1*). This grid is an output of the flow computation.
10. Buildup grid (*buildup*, Chapter 6). This grid contains the development rate assumed in each cell.
11. Water land use zones grid (*zone_gr*, Chapter 6). Water must be given the grid code 999.

    The remaining input files are required only if a recharge zone is considered.
12. Predicted total flow grid in cfs (*flow1*). This grid is computed with the flow computation grid (*Qual.Flow*).
13. Total flow without considering recharge in cfs (*tflow01*).
14. Cell recharge in cfs (*lcorr_rech*, Chapter 5).

- **Qual.BMPfut output files**

The script creates a new load grid for each constituent considered. If a recharge zone is considered and if non discharge BMPs (e.g. COMP, SOS, Chapter 7) are located within the study area, the script creates also a new flow grid (*newflo1*). This grid takes into account the volume of water removed by the non discharge BMPs (e.g. for BOD and COD in Figure 8.71).

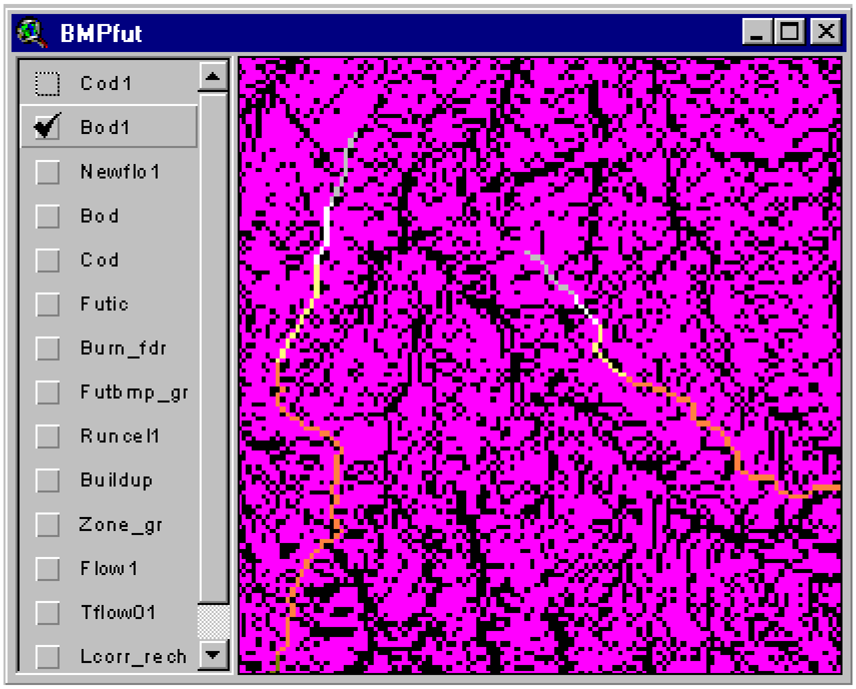


Figure 8.71: Inputs and outputs (Qual.BMPfut)

- **Format for input tables**

(a) Average capture volume table (Figure 8.72)

Average capture volumes are defined for each BMP. Linear relationships were established between impervious cover and average capture volume (Chapter 7, Table 7.2).

$$ACV = a * IC + b, \text{ with } 0<IC<1 \text{ and } 0<ACV<1.$$

Two parameters, a and b, define the relationship for each BMP. The average capture volume table is built so that each column corresponds to a BMP. The first row contains the value of the first order coefficient a and the second row the value of the constant b. The table has as many columns as average capture volume relationships needed (3 in the study).

acv.dbf

SED1-SAND2	SAND3-COMP	SOS
-0.4714	-0.1540	0.0000
0.9960	0.9762	1.0000

Figure 8.72: Average Capture Volume table (Qual.BMPfut)

(b) BMPs zones table (Figure 8.73)

A BMP zone is an area where a specific combination of BMPs is applied. Eight BMP zones were used in the study (Chapter 7, Table 7.1). The BMP zones table is built such as each zone corresponds to a row. The zone number is given in the first column (1 to 8 in this case). It corresponds to the grid-code of the cell located within the zone. Each following column corresponds to a type of BMP. The table gives the contribution of each BMP in each zone. For example, the first zone is treated by Sand2 (44%), COMP (36%) and SOS (remaining 20%).

bmpzone.dbf

Bmpzone	Sed1	Sand2	Sand3	Comp	Sos
1	0.00	0.44	0.00	0.36	0.20
2	0.00	1.00	0.00	0.00	0.00
3	0.10	0.34	0.00	0.36	0.20
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.17	0.10	0.00	0.00
6	0.00	0.24	0.62	0.00	0.00
7	0.00	0.44	0.56	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00

Figure 8.73: BMPs zones table (Qual.BMPfut)

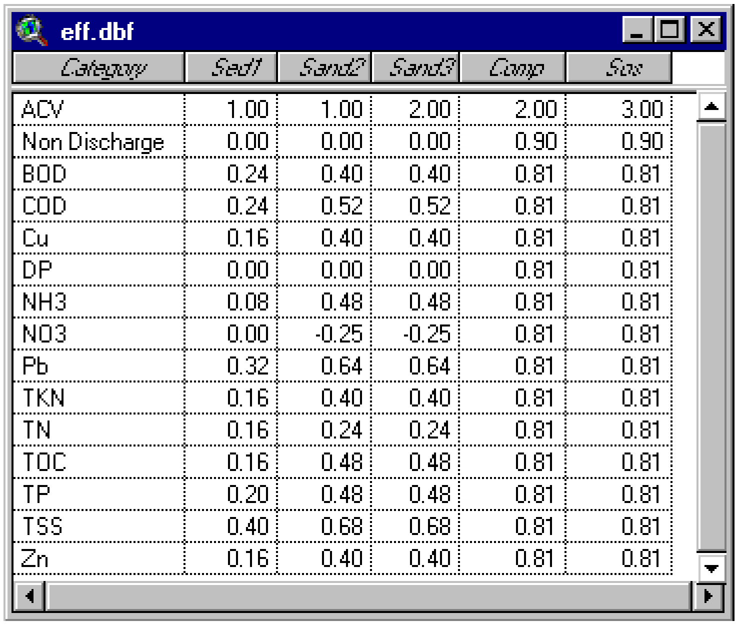
(c) Efficiency table (Figure 8.74)

The efficiency table defines the efficiencies of each BMP for each constituent. It indicates also the Average Capture Volume relationship for each BMP. Finally, it contains also the flow removal efficiency if the BMP is a non discharge BMP.

The first column indicates what the information in each row refers to (ACV, non discharge or a specific constituent). Each following column defines the characteristics of a type of BMP.

The first row (*ACV*) allows the program to determine the average capture volume relationship to apply to each BMP. The numbers refer to the column number in the average capture volume table. For example, the number 2 in the first row for the SAND3 column indicates that the coefficient given in the third column (the third column has for order 2 in the list of fields) of the average capture volume table will be use to compute the average capture volume for the SAND3 BMP. The second row (*non discharge*) contains the flow removal efficiency associated with each BMP. This efficiency is equal to zero, except for non discharge BMPs (e.g. COMP and SOS which have a 90% flow removal efficiency). The other rows contain the load removal efficiencies for the constituent whose name is in the first column.

Note that it is not necessary that the number of constituents or their order in the efficiency and in the direct runoff table be identical. However a constituent must have the same name in the two tables.



eff.dbf

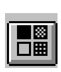
Category	Sed1	Sand2	Sand3	Comp	Sos
ACV	1.00	1.00	2.00	2.00	3.00
Non Discharge	0.00	0.00	0.00	0.90	0.90
BOD	0.24	0.40	0.40	0.81	0.81
COD	0.24	0.52	0.52	0.81	0.81
Cu	0.16	0.40	0.40	0.81	0.81
DP	0.00	0.00	0.00	0.81	0.81
NH3	0.08	0.48	0.48	0.81	0.81
NO3	0.00	-0.25	-0.25	0.81	0.81
Pb	0.32	0.64	0.64	0.81	0.81
TKN	0.16	0.40	0.40	0.81	0.81
TN	0.16	0.24	0.24	0.81	0.81
TOC	0.16	0.48	0.48	0.81	0.81
TP	0.20	0.48	0.48	0.81	0.81
TSS	0.40	0.68	0.68	0.81	0.81
Zn	0.16	0.40	0.40	0.81	0.81

Figure 8.74: Efficiency table (Qual.BMPfut)

(d) Direct runoff EMC table (Figure 8.32)

The direct runoff table is described in section 8.3.1. Its format must be consistent with the format of the efficiency table (same name). Each constituent given in the efficiency table must also be defined in the direct runoff table.

- **Running Qual.BMPfut**

Once all the inputs have been added to the project, the script can be run by clicking on the button  or by using the command *Qual/BMPfut*.

The user must first set the analysis extent and the cell size in the *Analysis Properties* window and then choose the working directory where the temporary files will be written (Figure 8.75).

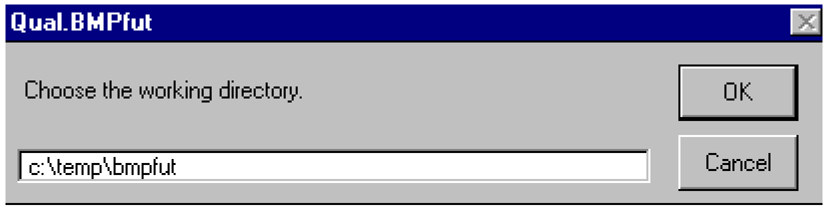


Figure 8.75: Choose a working directory (Qual.BMPfut)

The user is then asked whether he or she wants to consider a recharge zone (Figure 8.76).

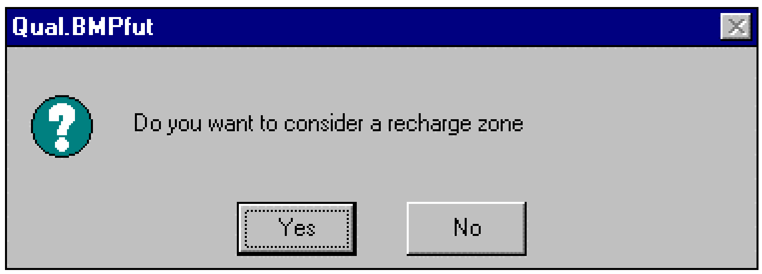


Figure 8.76: Consider a recharge zone (Qual.BMPfut)

The script prompts then the user for a series of tables, which must have been added previously to the project window. The first table needed is the average capture volume table (*acv.dbf*, Figure 8.77).

Figure 8.77: Choose an average capture volume table (Qual.BMPfut)

The following message box prompts the user for the BMP zones table which defines the combination of BMPs within each zone (*bmpzone.dbf*, Figure 8.78).

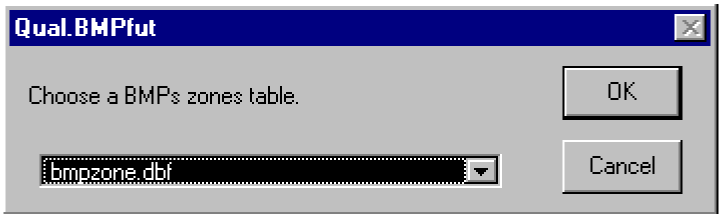


Figure 8.78: Choose a BMP zones table (Qual.BMPfut)

The efficiency table, defining the removal load efficiencies for each BMPs, is required next (*eff.dbf*, Figure 8.79).

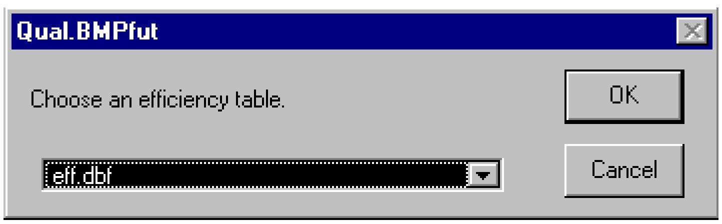


Figure 8.79: Choose an efficiency table (Qual.BMPfut)

The last input table is the direct runoff EMC table (*emcrun.dbf*, Figure 8.80).

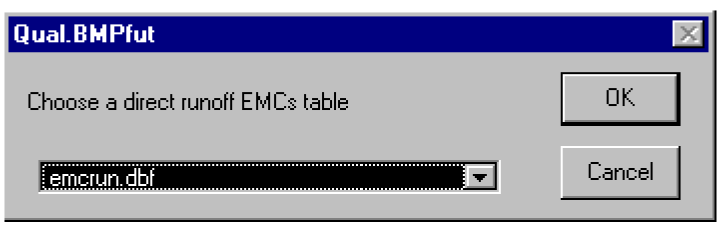


Figure 8.80: Choose a direct runoff EMC table (Qual.BMPfut)

The script uses then the efficiency table to create a list of the constituents for which the new load grid can be computed. It is necessary for the constituents to be also defined in the direct runoff EMC table. The user is prompted to choose the constituents to model (Figure 8.81).

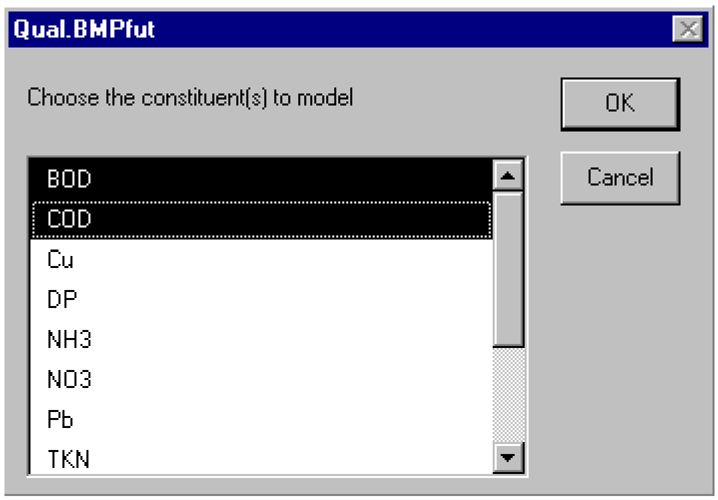


Figure 8.81: Choose the constituent(s) to model (Qual.BMPfut)

For each selected constituent, the user is prompted for the name to give to the new load (Figure 8.82) and for the initial load grid to use in the computations (Figure 8.83).

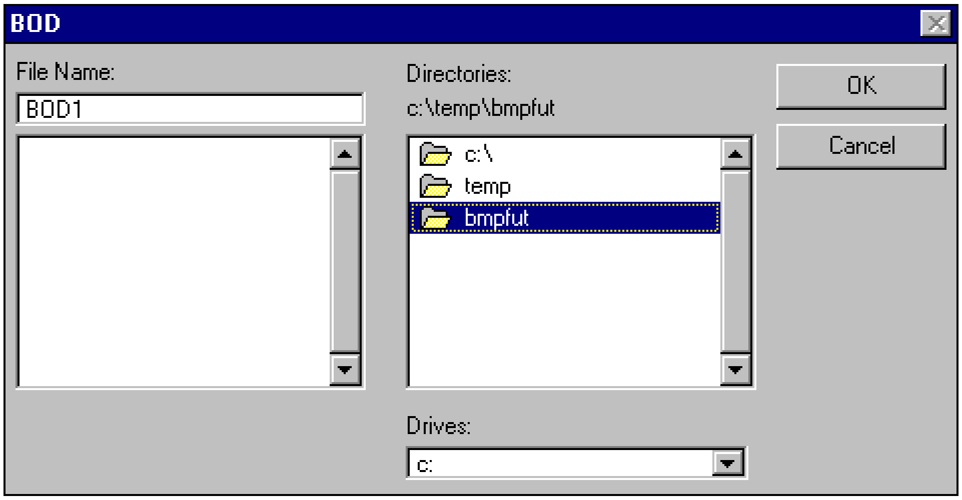


Figure 8.82: Name the output load grid for BOD (Qual.BMPfut)

Figure 8.83: Choose an initial load grid for BOD (Qual.BMPfut)

The program prompts then for a series of themes which must have been previously added to the view. The first theme required is an impervious cover theme (*futic*, Figure 8.84). It can be either a grid or a polygon coverage. If it is a polygon coverage, then another window prompts the user for the name of the impervious cover field.

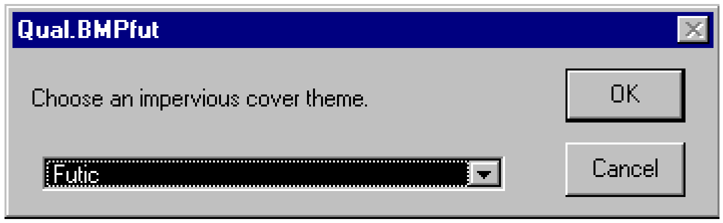


Figure 8.84: Choose an impervious cover theme (Qual.BMPfut)

The user is prompted next for the flowdirection grid (*burn_fdr*, Figure 8.85).

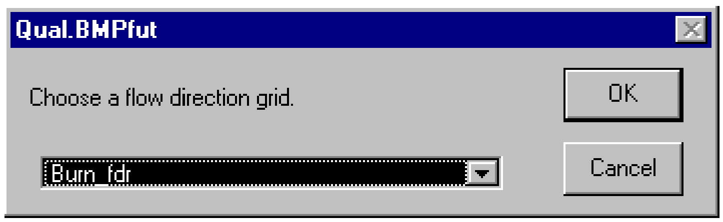


Figure 8.85: Choose a flow direction grid (Qual.BMPfut)

The following message box prompts the user for a BMP zones grid (*futbmp_gr*, Figure 8.86). This grid is used with the BMP zones table to characterize the combination of BMPs to apply to each cell.

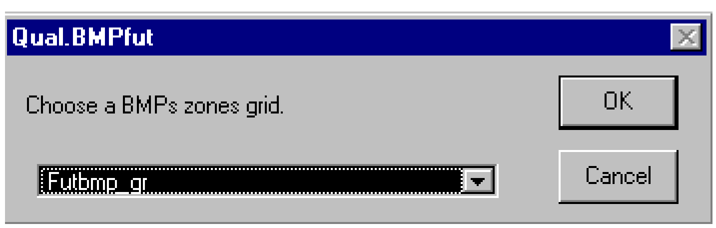


Figure 8.86: Choose a BMP zones grid (Qual.BMPfut)

The grid containing the corrected amount of runoff generated in each cell is then required (*runcel1*, Figure 8.87).

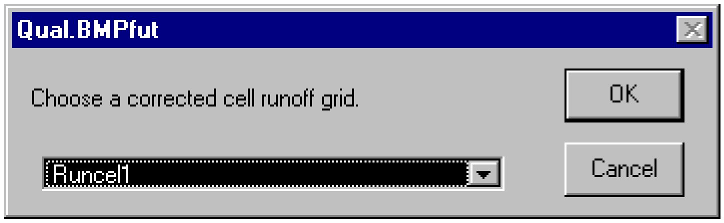


Figure 8.87: Choose a corrected cell runoff grid (Qual.BMPfut)

The BMPs treat only newly developed area. The development in each cell is given by a buildup grid (*buildup*, Figure 8.88) which contains the development averaged over the traffic serial zones (Chapter 6).

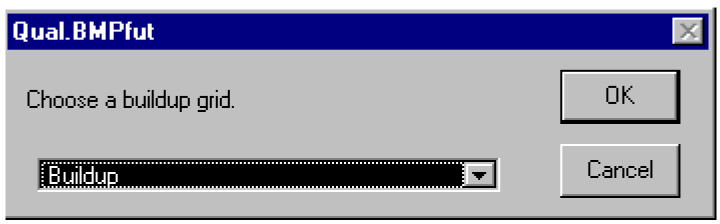


Figure 8.88: Choose a buildup grid (Qual.BMPfut)

As no pollution is generated in the water, a water land use grid (*zone_gr*, Figure 8.89) is required to set the EMCs in water land uses zones to zero. Water must be given the grid code 999, which corresponds to the water code used in the traffic serial zones.

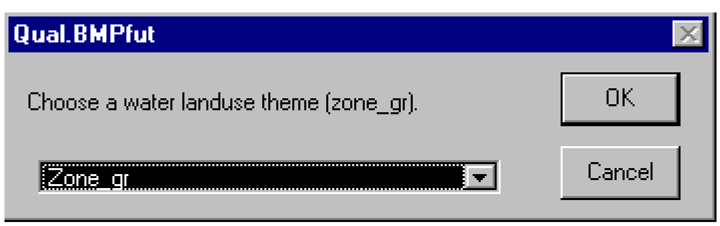


Figure 8.89: Choose a water land use theme (Qual.BMPfut)

The last three inputs are required only if a recharge zone is considered. They are used to modify the load lost to the recharge zone. The first input is a predicted flow grid (*flow1*, Figure 8.90). This grid is obtained by using the flow computation script.

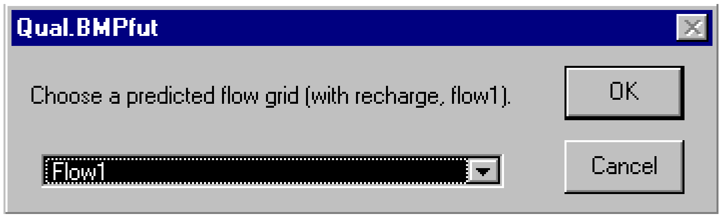


Figure 8.90: Choose a predicted flow grid (Qual.BMPfut)

The second input needed to deal with the recharge is a grid containing the total flow without considering the recharge (*tflow01*, Figure 8.91).

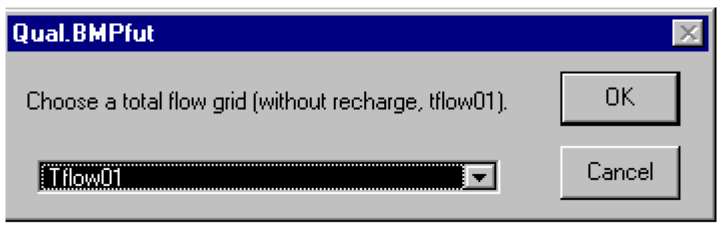


Figure 8.91: Choose a total flow grid without recharge (Qual.BMPfut)

Finally, the last input file is grid containing the flow lost to the recharge in each cell (*lcorr_rech*, Figure 8.92).

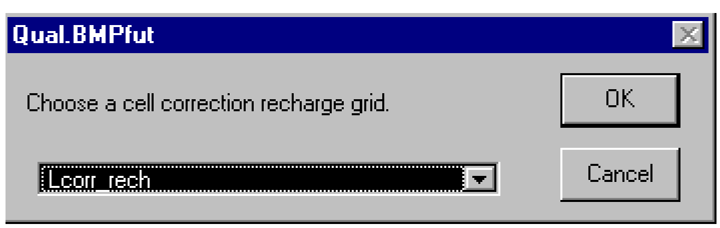


Figure 8.92: Choose a cell recharge grid (Qual.BMPfut)

The scripts uses all the input files to compute the new load(s) accordingly to the methodology presented in Chapter 7. A message box indicates to the user that the new grids have been calculated and that the computation is over (Figure 8.93).

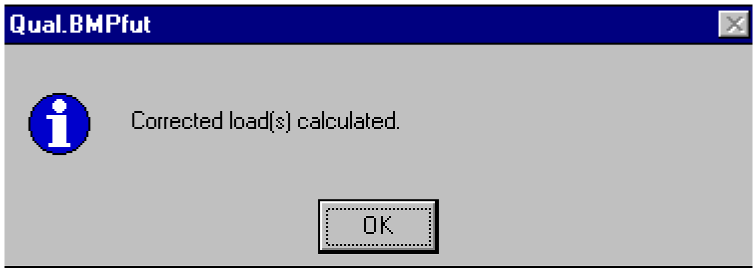


Figure 8.93: Final message to user (Qual.BMPfut)

A corrected load grid has been computed for each selected constituent (Figure 8.71). If non-discharge BMPs are considered, the script creates also a new flow grid (*newflow1*) which is the predicted flow (*flow1*) corrected with the flow removed by the non-discharge BMPs.

Chapter 9: Conclusions and Recommendations

The goal of this research project was to determine current and future non-point source pollution loads in Austin streams. Current flows matching observed flows at 17 USGS stations were determined in Arc/Info, and loads based on the flow were established for 122 sites (Environmental Integrity Index sites, USGS stations and mouths) within the study area. The effects of both located and non-located BMPs were modeled. The results can be found in the CDROM (Appendix B). The model was finally converted to be used in ArcView which possesses a better user interface than Arc/Info.

This GIS-based non-point source pollution study is composed of three major steps. First, the data characterizing the study area were gathered by monitoring, doing field observations or looking into databases (e.g. USGS). Then the impervious cover/runoff coefficient and impervious cover/event mean concentration relationships were established. The data used as input to the model (precipitation, impervious cover) or as references (observed discharges and concentrations) were derived from the observations available. Finally, these data and relationships were used in a GIS to compute the predicted flows and loads.

A 30m grid representation of the landscape composed of about 5 million cells was laid over Austin, and discharge and loads generated at each cell. *Weighted flowaccumulations were* done to sum the contribution of all upstream cells at each downstream cell. Resulting flows were calibrated to match observed values at gages, and calibration grids were created to adjust flows determined from regional relationships to allow for local flow variations in particular watersheds.

Land surface loads were computed by multiplying the flows generated in each cell by the associated event mean concentration, and by summing the contribution of all upstream cells at each downstream cell. In-stream loads due to erosion were assumed to be the difference between observed and predicted loads for current conditions. A relationship relating the erosion load per channel length to the watershed average impervious cover was established to extrapolate erosion to the watersheds for which no observed data were available.

The model was created and implemented for the Austin area. The same concept and much of the same code can however be used for any location for which the flowdirection grid can be computed and where a flow calibration can be done.

The principal difficulty in the Austin study was to model the recharge zone: variations in the loads coming into the recharge zone modify also the load lost to the recharge zone. Another problem arose from the fact that the *flowaccumulation* function does not handle negative values. Modifying the function so that it can deal with negative values would noticeably reduce the computation time.

An important difference between this study and the non-point source pollution studies previously done at CRWR was that the Austin study takes into account both direct runoff and base flow. As the underground topography was not known, base flow was assumed to have the same path as direct runoff, but different associated pollutant concentrations (EMC).

Most of the computations in this project were done in the GIS software Arc/Info and a discussion of the methodology was presented in Chapters 5 (flow), 6 (load) and 7

(BMPs). The advantage of Arc/Info over ArcView is a decreased computation time due to the capability to run several programs at the same time. However, the programs in Arc/Info are not very flexible, and the computing environment is not very user friendly.

As data vary rapidly, flexibility became a growing concern. The Spatial Analyst extension developed by ESRI offered the possibility to use grids in ArcView. At the beginning of the study, ArcView was principally used only to display data or to retrieve some information from the grids. Its use as a computation environment became more and more appealing as the number of variables to be modified grew (future BMP assumptions). ArcView also allowed the creation of a user-friendly environment employing menus and buttons.

The model aims at being as flexible as possible. The BMP parameters and the EMCs can be easily modified and they do not require the analyst to recalibrate the model (for the load, the erosion calibration is done in Excel). However, any modification of the current land use conditions, of the precipitation value used, or of the impervious cover/runoff coefficient relationships will require recalibration of the model. At present, the calibration was not part of the ArcView version of the model. However, the need for a totally integrated model in ArcView leads to the need to transpose the calibration process from Arc/Info to ArcView. In the study, a flow correction coefficient grid (*corcoef*) was defined for current conditions and the same grid was applied for future conditions. This grid was created by using correction coefficients obtained directly from comparisons between observed and predicted flow and by applying a standard relationship at ungaged sites. The extrapolation formula relates linearly the flow correction coefficient to the subwatershed average impervious cover. Since the impervious cover is modified for future conditions, so will the correction coefficients based on the equation. The user may decide to modify the correction grid according to the variations in the impervious cover.

The ultimate objective was to create a basic tool which can be further developed as knowledge and information concerning non-point source pollution increase. Further development of the model should be closely related to the feedback from the users. Close interactions with the City of Austin have allowed the model to be modified according to the City's expectations. Further scripts can be easily derived from the five model scripts developed in Chapter 8. Other programs were developed for the study (separation between direct runoff and base flow loads, load per jurisdiction type…). There are not presented in this report as they are just a variation on a general theme.

A principal innovation in this research is the study of the effects of Best Management Practices on water quality in a GIS environment. Three different scenarios were considered:

- Located BMPs defined by load removed.
- Located BMPs defined by efficiency.
- Non located BMPs defined by efficiency.

The two methods dealing with located BMPs rely on the *flowaccumulation* function, since each BMP modifies the load downstream of its location. The second approach was not implemented since the City of Austin supplied the loads removed for the located BMPs. However it has also been very instructive to develop the more complex method for located BMPs defined with efficiency. It was this method which triggered the conversion from Arc/Info to ArcView which offers much more flexibility in programming, and especially the use of loops programming.

The main problem to solve for future conditions is that the exact location of the BMPs is not known. The solution is to assume that the effect of the BMPs is not concentrated in one location but is diffuse over an area. The analogy with located BMPs would be, instead of considering that the load is removed at the BMP, to assume that the load is removed in each cell where it is generated within the drainage area. The difference being that for non located BMPs both BMP location and drainage area are not known. It must be noted also that, since the relationships used to compute the load removed are nonlinear, computing the load removed in each cell and then summing over all the cells or computing the total load removed in the BMP cell will yield different results.

Future work should concentrate on the use of finer elevation data (orthophotos) and on the study of the elements which modify the flow path (sewer network). The model gives the basis of a watershed based water quality non point source pollution analysis. It should be transformed with the different needs of the users.

APPENDIX A

DATA DICTIONARY

The data dictionary contains the data used and created in the study in alphabetical order. The data considered are coverages, shapefiles, grids, tables, Arcview project and extension. The scripts are presented in Appendices B and C, and are not considered in the data dictionary. The following conventions are used:

[pC]	Point Coverage
[AC]	Arc Coverage
[PC]	Polygon Coverage
[Gr]	Grid
[I]	Image
[Table]	Table
[Project]	Avenue Project
[Ext]	Avenue Extension

All coverages and grids are in State Plane coordinates, unless otherwise specified.

Name	Feature	Class	Attribute	Value	Description
Acv.dbf	Annual capture volume table.	[Table]			This table is an input to the script Qual.BMPfut (non located BMPs).
Austin.tif	1:24,000 scanned map for Austin region	[I]			This map results from the projection in state plane coordinates of a scanned map in geographic coordinates.
Austn_e	30m DEM for Austin East. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Austn_w	30m DEM for Austin East. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level.
Bastrpsw	30m DEM for Bastrop SW. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level.
Beecave	30m DEM for Bee Cave. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level.
Bmpzone.dbf	BMP zones table.	[Table]			This table is an input to the script Qual.BMPfut (non located BMPs).
Border	Border of the study area.	[PC]			This coverage defines a drainage area.
Buda	30m DEM for Buda. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level.
Buildup	%of zones under construction	[Gr]		Floating point	This grid is used in the construction load and in the future BMPs calibration.
Burn_crk	Digitized creeks coverage from COA (creeks) completed with the EPA Reach Files 3 (RF3)	[AC]			

Name	Feature	Class	Attribute	Value	Description
Burn_dem	Original DEM raised by 10,000 ft and with the streams burnt in	[Gr]	elevation	Floating point	Elevation in ft (original elevation + 10,000 ft except in the streams)
Burn_fac	Flow accumulation grid	[Gr]		Floating point	The grid code for each cell corresponds to the number of cells upstream of this cell.
Burn_fdr	Flow direction grid	[Gr]		integer	The grid code shows which of the 8 neighboring cells lies on the path of steepest descet. The 8 directions are labelled as 1(E), 2(SE), 4(S), 8(SW), 16(W), 32(NW), 64(N) and 128(NE).
Burn_fil	Filled grid	[Gr]		Floating point	
Canyon_pt	Barton canyon points	[pC]			
Canyon_wshd	Barton canyon wshd				
City1	Current residential BMPs	[pC]	Current loads removed		121 BMPs.
Cityf1	Future residential BMPs	[pC]	Future loads removed		121 BMPs. This coverage is identical to city1. Only the attribute table is different.
Coawshd_cv	COA watersheds delineated from the 30m DEMs.	[PC]	name		45 watersheds
Coawshd_gr	COA watersheds delineated from the 30m DEMs	[Gr]		integer	45 watersheds

Name	Feature	Class	Attribute	Value	Description
Coawshd_pt	COA watersheds outlets	[pC]			This coverage is use to find the average characteristics of the watersheds.
Com1pd27	Commercial BMPs	[pC]			229 BMPs. Current and future loads removed are the same.
Corcoef	Flow correction coefficients used in the calibration process.	[Gr]		Floating point	
Coupland	30m DEM for Coupland. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Creedmor	30m DEM for Creedmoor. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Creeks	Principal streams of the Austin area	[AC]			Source: COA
Crk100_cv	Delineated streams (threshold 100 cells)	[AC]			
Crk100_gr	Delineated streams (threshold 100 cells)	[Gr]			
Crk1k_cv	Delineated streams (threshold 1,000 cells)	[AC]			
Crk1k_gr	Delineated streams (threshold 100 cells)	[Gr]			
Crk10k_cv	Delineated streams (threshold 10,000 cells)	[AC]			
Crk10k_gr	Delineated streams (threshold 100 cells)	[Gr]			

Name	Feature	Class	Attribute	Value	Description
Crk500_cv	Delineated streams (threshold 500 cells)	[AC]			
Crk500_gr	Delineated streams (threshold 100 cells)	[Gr]			
Crk5k_cv	Delineated streams (threshold 5,000 cells)	[AC]			
Crk5k_gr	Delineated streams (threshold 100 cells)	[Gr]			
Dbl1	Double line roads for North Austin	[AC]			Source: COA
Dbl2	Double line roads for South Austin	[AC]			Source: COA
Dem_st	Merged and projected grid of the 27 30m DEM's	[Gr]	elevation	Floating point	The elevation values in each grid-cell are in units of feet above sea level.
Drift	30m DEM for Driftwood. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Dripsprg	30m DEM for Dripping Spring. UTM projection	[Gr]	Elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Edwards	Edwards aquifer recharge and contributing zones	[PC]	Zone_name	Artesian Artesian (non potable) Contributing Excluded Recharge Transition USGS recharge	Source: Barton Springs Conservation District

Name	Feature	Class	Attribute	Value	Description
Edw_tnris	Edwards aquifer	[PC]			Source: TNRIS
Eff.dbf	Non located BMP efficiency table.	[Table]			This table is an input to the script Qual.BMPfut (non located BMPs).
EII	EII sites	[pC]			
EII_wshd	EII sites watersheds	[PC]			
Elgin_w	30m DEM for Elgin West. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Emcbf.dbf	Base flow EMC table.	[Table]			This table is an input to the script Qual.load (load computation).
Emcrun.dbf	Direct runoff EMC table.	[Table]			This table is an input to the scripts Qual.Load (load computation) and Qual.BMPfut (non located BMPs).

Name	Feature	Class	Attribute	Value	Description
EPA_luse	Land use of the Austin region according to the USGS.	[PC]	lucode	0	Unknown
				11	Residential
				12	Commercial services
				13	Industrial
				14	Transportation, communication
				15	Industrial and commercial
				16	Mixed urban or build-up land
				17	Other urban or build-up land
				21	Cropland and pasture
				22	Orchards, groves, vineyards, nurseries
				23	Confined feeding operations
				24	Other agricultural
				31	Herbaceous rangeland
				32	Shrub and brush rangeland
				33	Mixed rangeland
				41	Desidous forest land
				42	Evergreen forest land
				43	Mixed forest land
				51	Streams and canals
				52	Lakes
				53	Reservoirs
				54	Bays and estuaries
				61	Forestred wetlands
				62	Nonforestred wetlands
				71	Dry salt flat
				72	Beaches
				73	Sandy areas other than beach
				74	Bare exposed rock
				75	Strip mines, quarries and gravel Pits
				76	Transitional areas
				77	Mixed barren lands

Name	Feature	Class	Attribute	Value	Description
Er_pt	Erosion points.	[pC]			
Fin_luse	Current land use based on the COA land use (land use) completed with the USGS land use (EPA_luse).	[PC]	Newlanduse code	100, 113 200 300 400 500, 560 600 700 800, 870 900, 999 940	Single family Multi-family Commercial Office Industrial Civic/educational Park Transportation Vacant/undeveloped Water/lake
Futbmp_cv	Non located BMPs zones.	[PC]			
Futvbmp_gr	Non located BMPs zones.	[Gr]			
Futic	Future impervious cover	[Gr]			
Future1	Retrofit BMPs for future conditions only	[pC]			11 BMPs.
Hamcross	30m DEM for Hammetts Crossing. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Henly	30m DEM for Henly. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Inlets	Storm sewers inlets.	[pC]			Source: COA
Jollyvil	30m DEM for Jollyville. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Juris	Jurisdictions	[PC]			
Lakes	Lakes of the Austin area	[PC]			

Name	Feature	Class	Attribute	Value	Description
Landuse	1990 landuse of the Austin region according to COA	[PC]	Lucode	100	Single family
				113	Mobile home
				200	Multi-family
				300	Commercial
				400	Office
				500	Industrial
				560	Open extraction
				600	Civic/educational
				700	Park
				800	Transportation
				870	Utilities
				900	Vacant/undeveloped
				940	Water/lake
				999	Unknown
Lcorr_rech	Cell recharge (cfs).	[Gr]			
Length	Flow length in feet based on the burnt in creeks.	[Gr]			
Load.xls	Excel spreadsheet with current future and construction loads computed in the study.	[Table]			
Lyttsprf	30m DEM for Lytton Springs. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Manfiel	30m DEM for Mansfield. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Manor	30m DEM for Manor. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Monto	30m DEM for Montopolis. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level

Name	Feature	Class	Attribute	Value	Description
Mountcty	30m DEM for Mountain City. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Newsites	All City of Austin drainage utility sampling sites (COA).	[pC]			Source: COA
Oakhill	30m DEM for Oak Hill. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Pflug_e	30m DEM for Pflugerville East. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Pflug_w	30m DEM for Pflugerville West. UTM projection	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Pt64_cv	Upstream limit of the 64 acres creeks.	[pC]			
Qual.apr	Non-point source pollution ArcView project	[Project]			This project was used to build the water quality extension.
Qual.avx	Non-point source pollution extension.	[Ext]			
Recharge	Recharge zone of the Edwards aquifer	[PC]	zone	North Central South	This coverage was used in the study.
Rech_fac	Recharge flow (in cfs).	[Gr]			
Roads	Principal streets and roads of the Austin area.	[AC]			Source: COA
Roughhol	30m DEM for Rough Holllow. UTM projection	[Gr]	Elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level

Name	Feature	Class	Attribute	Value	Description
Runcoef	Runoff coefficient grid for the USGS stations.	[Gr]			Used as input of flow.aml for the calibration.
Shinhhi	30m DEM for Shingle Hills. UTM projection.	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Signalhi	30m DEM for Signal Hill. UTM projection.	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Stations	Sites studied in the project.	[pC]			This point coverage contains the points located at the EII sites, at the USGS stations and at the mouths.
Stormsew	Storm sewers.	[AC]			Source: COA
Texas	Texas counties.	[PC]	name		
USGS	USGS stations used as references for model calibration	[pC]	Observed flow (cfs)		17 stations.
USGS_quad	USGS 1:24,000 quadrangles corresponding to the DEMs used in the project.	[PC]	Name		
USGS_wshd	USGS stations watersheds	[PC]			
Utley	30m DEM for Utley. UTM projection.	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Water_gr	Merged streams and lakes at their original elevation value.	[Gr]	elevation	Floating point	The elevation values in each grid-cell are in units of feet above sea level.
Webber	30m DEM for Webberville. UTM projection.	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level
Wqexempt27	Water quality exempt zones.	[PC]			Source: COA
Wqpza27	Water quality protection zones	[PC]			Source: COA

Name	Feature	Class	Attribute	Value	Description
Wsheds	Watersheds of the Austin area	[PC]	name		Source: COA
Wshd1k_cv	Delineated watersheds of the Austin area (threshold 1,000 cells).	[PC]			
Wshd1k_gr	Delineated watersheds of the Austin area (threshold 1,000 cells).	[Gr]			
Wshd10k_cv	Delineated watersheds of the Austin area (threshold 10,000 cells).	[PC]			
Wshd10k_gr	Delineated watersheds of the Austin area (threshold 10,000 cells).	[Gr]			
Wshd500_cv	Delineated watersheds of the Austin area (threshold 500 cells).	[PC]			
Wshd500_gr	Delineated watersheds of the Austin area (threshold 500 cells).	[Gr]			
Wshd5k_cv	Delineated watersheds of the Austin area (threshold 5,000 cells).	[PC]			
Wshd5k_gr	Delineated watersheds of the Austin area (threshold 5,000 cells).	[Gr]			
Yeager	30m DEM for Yeager Creek. UTM projection.	[Gr]	elevation	Floating point	The elevation values in each grid cell are in units of feet above sea level

Name	Feature	Class	Attribute	Value	Description
Zones	Traffic serial zones	[PC]		integer	This coverage is used to determine future impervious cover and future land use..
205_st	EPA Reach File 3 (RF3) 12090205	[AC]			The ID number corresponds to the Hydrologic Unit Code (HUC).
301_st	EPA Reach File 3 (RF3) 12090301	[AC]			The ID number corresponds to the Hydrologic Unit Code (HUC).
64crk	Delineated streams (threshold 64 acres)	[AC]			

APPENDIX B

CDROM DOCUMENTATION

CDROM Documentation

The CDROM contains the same data as the data dictionary but also the Avenue scripts. The data are not classified by alphabetical order but by theme.All coverages and grids are in Texas State Plane coordinates except otherwise specified.

COORDINATE SYSTEM DESCRIPTION

Projection	STATEPLANE		
Zone	5376		
Datum	NAD27		
Units	FEET	Spheroid	CLARKE1866
Parameters:			

The following listing describes the contents of each section of the CDROM. The following conventions are used :

[pC] Point Coverage
[AC] Arc Coverage
[PC] Polygon Coverage
[Gr] Grid
[I] Image
[Table] Table
[Script] Avenue Script
[Project] Avenue Project
[Ext] Avenue Extension

The source is given after the definition of the data.
ex : **creeks** [AC] principal streams in the Austin region (COA).

COA stands for City Of Austin.

Directories

- COA_DATA: coverages from external sources.
- DEM: DEMs and data derived from it.
- MODEL: programs and data used to run the model

DIRECTORY COA_DATA

- **bmp**
  - **city1** [pC] Current residential BMPs with loads removed (COA).
  - **cityf1** [pC] Future residential BMPs (same coverage as city1 but different values for the loads removed) (COA).
  - **com1pd27** [pC] Commercial BMPs (same values for current and future conditions).
  - **future1** [pC] Retrofit BMPs (future conditions only).

- **edwards**
  - **edw_tnris** [PC] Edwards aquifer (Texas Natural Resources Information System).
  - **edwards** [PC] Edwards aquifer recharge and contributing zones in Austin (Barton Springs Conservation District).
  - **recharge** [PC] recharge zone coverage used in the study.

- **landuse**
  - **epa_luse** [PC] land use coverage used to complete the city coverage (EPA).
  - **fin_luse [**PC] completed current land use used in the study for current conditions.
  - **landuse** [PC] existing land use of the city (1990 version (COA).
  - **zones** [PC] Traffic serial zones (COA). The future impervious cover and land use are based on that coverage.

- **political**
  - **map**
    Both of the following files are needed to properly display the map.
    - **austinst.tif** [I] USGS 1:24,000 scanned maps of Austin in state plane coordinates. This map was scanned and the resulting image in geographic coordinates was projected in state plane coordinates (Horizons Technologies, Inc.).
    - **austin.tfw** displays information such as size, localization
  - **juris** [PC] jurisdiction coverage (COA).
  - **texas** [PC] Texas counties (TNRIS).
  - **usgs_quad** [PC] USGS quadrangles for the study area.
  - **wqexempt27** [PC] Water quality exempt zones (COA).
  - **wqpza27** [PC] Water quality protection zones (COA).
  - **wsheds** [PC] principal watersheds of Austin. They are named under the attribute Name (COA).

- **road**
    - **dbl1 & dbl2** [AC]double line street files. dbl1 covers the northern part of the city and dbl2 the southern part (COA).
    - **roads** [AC] principal streets and roads of Austin (COA).

- **sewers**
    - **inlets** [pC] storm sewer inlets (COA).
    - **stormsew** [AC] storm sewers (COA).

- **stations**
    - **64acres**
        - **64crk** [AC] streams (64 acres threshold).
        - **pt64_cv** [pC] upstream limit of the 64 acres creeks.
    - **canyon**
        - **canyon_pt** [pC] Barton canyon points.
        - **canyon_wshd** [PC] Barton canyon watersheds.
    - **coawshd_pt** [pC] City of Austin delineated watersheds (coawshd_cv) outlets.
    - **eii**
        - **eii** [pC] Environmental Integrity Index sites.
        - **eii_wshd** [PC] Environmental Integrity Index watersheds.
    - **er_pt** [pC] Erosion points.
    - **newsites** [pC] all City of Austin drainage utility sampling sites (COA).
    - **stations** [pC] sites studied. This point coverage contains the EII sites, the USGS stations and the creeks mouths where flows and loads have been determined in the study.
    - **usgs**
        - **usgs** [pC] USGS stations in Austin region with a period of record of at least one year.
        - **usgs_wshd** [PC] USGS watersheds.

- **streams**
    - **creeks** [AC] principal streams in the Austin Region (COA).
    - **lakes** [PC] lakes of Austin (COA).
    - **rf3**
        - **205_st** [AC] EPA reach file 3 for HUC (hydrologic unit code) 12090205 (EPA).
        - **301_st** [AC] EPA reach file 3 for HUC 12090301 (EPA).

DIRECTORY DEM (30M CELL SIZE)

This directory contains the inputs and outputs of the burn-in process and of the watershed and stream delineation.

- **border** [PC] border of the Austin region.

- **burn_crk** [AC] used to build the creeks grid (WATER_GR) used in the burn-in process. The coverage supplied by the City of Austin has been completed by the EPA RF3 coverage.

- **burn_dem** [Gr] : This DEM was built by adding 10,000 ft to DEM_ST and by merging the resulting grid with the grid water_gr.

- **burn_fac** [Gr] flow accumulation grid (based on burn_dem).

- **burn_fdr** [Gr] flow direction grid (based on burn_dem).

- **burn_fil** [Gr] burn_dem whose sinks have been filled.

- **coa_wshd** [PC] principal watersheds. It can be compared with the coverage given by the city.
    - **coawshd_cv** [PC] delineated watersheds.
    - **coawshd_gr** [Gr] delineated watersheds. The grid code represents the coawshd_cv-id.

- **deln_crk**

This directory contains grids and coverages which were created by applying the Arc/Info Grid tools to the 30m Digital Elevation Model covering the Austin area.

- **crk100_cv** [AC] streams (100 cells threshold).
- **crk100_gr** [Gr] streams (100 cells threshold).
- **crk10k_cv** [AC] streams (10,000 cells threshold).
- **crk10k_gr** [Gr] streams (10,000 cells threshold).
- **crk1k_cv** [AC] streams (1,000 cells threshold).
- **crk1k_gr** [Gr] streams (1,000 cells threshold).
- **crk500_cv** [AC] streams (500 cells threshold).
- **crk500_gr** [Gr] streams (500 cells threshold).
- **crk5k_cv**[AC] streams (5,000 cells threshold).
- **crk5k_gr** [Gr]streams (5,000 cells threshold).

♦ **deln_wshd**

- **wshd10k_cv** [PC] watersheds (10,000 cells threshold).
- **wshd10k_gr** [Gr] watersheds (10,000 cells threshold).
- **wshd1k_cv** [PC] watersheds (1,000 cells threshold).
- **wshd1k_gr** [Gr] watersheds (1,000 cells threshold).
- **wshd500_cv** [PC] watersheds (500 cells threshold).
- **wshd500_gr** [Gr] watersheds (500 cells threshold).
- **wshd5k_cv** [PC] watersheds (5,000 cells threshold).
- **wshd5k_gr** [Gr] watersheds (5,000 cells threshold).

♦ **dem30m**

This directory contains the original 30m DEMs (27) received from Vernon F. Meyer and Associates, Inc which have ben processed towards grids. The DEMs are in the UTM projection system. The elevation values in each grid-cell are in units of feet above sea level.

- **Austn_e** [Gr] Austin East.
- **Austn_w** [Gr] Austn_w.
- **Bastrpsw** [Gr] Bastrop SW.
- **Beecave** [Gr] Beecave.
- **Buda** [Gr] Buda.
- **Coupland** [Gr] Coupland.
- **Creedmor** [Gr] Creedmoor.
- **Drift** [Gr] Driftwood.
- **Dripsprg** [Gr] Dripping Springs.
- **Elgin_w** [Gr] Elgin West.
- **Hamcross** [Gr] Hammetts Crossing.
- **Henly** [Gr] Henly.
- **Jollyvil** [Gr] Jollyville.
- **Lyttsprf** [Gr] Lytton Spring.
- **Manfie**l [Gr] Manfiel.
- **Manor** [Gr] Manor.
- **Monto** [Gr] Montopolis.
- **Mountcty** [Gr] Mountain City.
- **Oakhill** [Gr] Oak Hill.
- **Pflug_e** [Gr] Pflugerville East.
- **Pflug_w** [Gr] Pflugerville West.
- **Roughol** [Gr] Rough Hollow.
- **Shinhhi** [Gr] Shingle Hills.
- **Signalhi** [Gr] Signal Hill.
- **Utley** [Gr] Utley.
- **Webber** [Gr] Webberville.
- **Yeager** [Gr] Yeager Creek.

- **dem_st** [Gr] this DEM was created by merging 27 30m Digital Elevation Models and by projecting the resulting grid in state plane coordinates (Vernon F. Meyer and Associates, Inc.).

- **water_gr** [Gr] creeks grid used in the burn-in process.

DIRECTORY MODEL

- **buildup** [Gr] buildup grid.

- **burn_fac** [Gr] flowaccumulation grid.

- **burn_fdr** [Gr] flowdirection grid.

- **corcoef** [Gr] flow correction grid.

- **fin_luse [**PC] current land use coverage (COA).

- **futic** [Gr] future impervious cover.

- **Length** [Gr] Flow length in feet grid based on the creeks burnt in.

- **lcorr_rech** [Gr] cell recharge.

- **load.xls** [Table] Excel spreadsheet containing current, future and construction loads computed in the study at the locations of the points in the point coverage stations.

- **futbmp_gr** [Gr] non located BMP zones.

- **futbmp_cv** [PC] non located BMP zones.

- **qual.apr** [Project] ArcView project used to create the non-point source pollution extension.

- **qual.avx** [Ext] Non-point source pollution extension.

- **rech_fac** [Gr] recharge flow.

- **Scripts**
  - **Ext_inst.ave** [Script] Extension.Install.
  - **Ext_make.ave** [Script] Extension.Make.
  - **Ext_uninst.ave** [Script] Extension.Uninstall.
  - **Qual_add.ave** [Script] Qual.Addpoint.
  - **Qual_avg.ave** [Script] Qual.Average.
  - **Qual_bmpeff .ave** [Script] Qual.BMPeff.
  - **Qual_bmpeffc_ave** [Script] Qual.BMPeffcpt.
  - **Qual_bmpfut.ave** [Script] Qual.BMPfut.
  - **Qual_bmpload.ave** [Script] Qual.BMPload.
  - **Qual_creek.ave** [Script] Qual.Creek.
  - **Qual_creeklim.ave** [Script] Qual.Creeklim.
  - **Qual_del.ave** [Script] Qual.Delete.
  - **Qual_flow.ave** [Script] Qual.Flow.
  - **Qual_hydro.ave** [Script] Qual.HydroZdlsv.
  - **Qual_join.ave** [Script] Qual.Join.
  - **Qual_load.ave** [Script] Qual.Load.
  - **Qual_merge.ave** [Script] Qual.Mergetheme.
  - **Qual_pick.ave** [Script] Qual.Pick.
  - **Qual_wshd.ave** [Script] Qual.Wshd.
  - **Qual_zone.ave** [Script] Qual.Zonalmean.

- **tables**
  - **acv.dbf** [table] annual capture volume table.
  - **bmpzone** [table] BMP zones table.
  - **eff.dbf** [table] non located BMP efficiency table.
  - **emcbf.dbf** [table] base flow EMC table.
  - **emcrun.dbf** [table] direct runoff EMC table.

- **wshdzone** [Gr] watershed zones grid used to define the nested watersheds. watersheds)

- **zone_gr** [Gr] traffic serial zone grid used to locate water land use (grid-code 999).

APPENDIX C

AVENUE SCRIPTS

SCRIPTS

Script Name	Function
Extension.Install	Installs the extension.
Extension.Make	Creates the extension file qual.avx.
Extension.Uninstall	Uninstalls the extension.
Qual.Addpoint	Creates or adds a point to a point coverage.
Qual.Average	Computes the average based on drainage area.
Qual.BMPeff	Computes the load removed by located BMPs defined by removal efficiency.
Qual.BMPeff_cpt	Subroutine to BMP_eff.
Qual.BMPfut	Computes the new load after the effect of non located BMPs defined by removal efficiency.
Qual.BMPload	Computes the load removed by located BMPs defined by load removed.
Qual.Creek	Delineates creeks for a flowaccumulation threshold.
Qual.Creeklimit	Creates gridpoints representing the upstream limit of the creeks for a flowaccumulation threshold.
Qual.Delete	Deletes several fields in a table.
Qual.Flow	Computes the discharge (cfs).
Qual.HydroZdlsv	Dangling polygons.
Qual.Join	Permanently joins two tables.
Qual.Load	Computes the load (kg/yr).
Qual.Mergetheme	Merges two themes.
Qual.Pick	Retrieves the values of grids for points in a point coverage.
Qual.Wshd	Delineates watersheds for points in a point coverage.
Qual.Zonalmean	Computes the averages within a zone.

'Script: Extension.Install

```
'----------------------------
'--- Creation information ---
'----------------------------

'Name: Extension.Uninstall
'Version: 1.0
'Creation date: 11/17/97
'Author: Christine Dartiguenave
'Center for Research in Water Resources
'The University of Texas at Austin
'darti@crwr.utexas.edu


'---------------------------
'--- Purpose/Description ---
'---------------------------

'Uninstall the water quality extension.


'The self object for this script is the extension.
'The first object is the view, then the button and the menu

'Check if there is an active project

  if (av.getproject = nil) then
   return nil
  end

'Add the buttons at the end

thebuttonbar = av.getproject.findgui("View").getbuttonbar

thebutton1 = thebuttonbar.findbyname("Qual.Flow")
  if (thebutton1 = nil) then
    thebuttonbar.add(self.get(0),999)
  end

thebutton2 = thebuttonbar.findbyscript("Qual.Load")
  if (thebutton2 = nil) then
    thebuttonbar.add(self.get(1),999)
  end

thebutton3 = thebuttonbar.findbyscript("Qual.BMPload")
  if (thebutton3 =nil) then
    thebuttonbar.add(self.get(2),999)
  end

thebutton4 = thebuttonbar.findbyscript("Qual.BMPeff")
  if (thebutton4 =nil) then
    thebuttonbar.add(self.get(3),999)
  end

thebutton6 = thebuttonbar.findbyscript("Qual.BMPfut")
  if (thebutton6 =nil) then
    thebuttonbar.add(self.get(4),999)
  end

thebutton7 = thebuttonbar.findbyscript("Qual.Average")
  if (thebutton7 =nil) then
    thebuttonbar.add(self.get(5),999)
  end

thebutton8 = thebuttonbar.findbyscript("Qual.Zonalmean")
  if (thebutton8 =nil) then
    thebuttonbar.add(self.get(6),999)
  end

thebutton9 = thebuttonbar.findbyscript("Qual.Wshd")
  if (thebutton9 =nil) then
    thebuttonbar.add(self.get(7),999)
  end

thebutton10 = thebuttonbar.findbyscript("Qual.Creek")
  if (thebutton10 =nil) then
    thebuttonbar.add(self.get(8),999)
  end

thebutton11 = thebuttonbar.findbyscript("Qual.Creeklimit")
  if (thebutton11 =nil) then
    thebuttonbar.add(self.get(9),999)
  end

thebutton12 = thebuttonbar.findbyscript("Qual.Pick")
  if (thebutton12 =nil) then
    thebuttonbar.add(self.get(10),999)
  end

thebutton13 = thebuttonbar.findbyscript("Qual.Mergetheme")
  if (thebutton13 =nil) then
    thebuttonbar.add(self.get(11),999)
  end

thebutton14 = thebuttonbar.findbyscript("Qual.HydroZdlsv")
  if (thebutton14 =nil) then
    thebuttonbar.add(self.get(12),999)
  end
```

```
thebuttonbar1 = av.getproject.findgui("Table").getbuttonbar

thebutton15 = thebuttonbar1.findbyscript("Qual.Delete")
  if (thebutton13 =nil) then
    thebuttonbar1.add(self.get(13),999)
  end

thebutton16 = thebuttonbar1.findbyscript("Qual.Join")
  if (thebutton14 =nil) then
    thebuttonbar1.add(self.get(14),999)
  end


'Add the tool

thetoolbar = av.getproject.findgui("view").gettoolbar
thetool = thetoolbar.findbyscript("qual.addpoint")
  if (thetool =nil) then
    thetoolbar.add(self.get(15),999)
  end

'Add the menus
themenubar = av.getproject.findgui("View").getmenubar

themenu1 = themenubar.findbylabel("Qual")
  if (themenu1 =nil) then
    themenubar.add(self.get(16),999)
  end

themenu2 = themenubar.findbylabel("QualTools")
  if (themenu2 =nil) then
    themenubar.add(self.get(17),999)
  end

themenubar2 = av.getproject.findgui("Table").getmenubar
themenu3 = themenubar2.findbylabel("QualTable")
  if (themenu3=nil) then
    themenubar2.add(self.get(18),999)
  end
```

'Script: Extension.Make

```
'----------------------------
'--- Creation information ---
'----------------------------

'Name: Extension.make
'Version: 1.0
'Creation date: 11/17/97
'Author: Christine Dartiguenave
'Center for Research in Water Resources
'The University of Texas at Austin
'darti@crwr.utexas.edu


'---------------------------
'--- Purpose/Description ---
'---------------------------

'Create the water quality extension


theproject=av.getproject

'install and uninstall scripts
theinstallscript=theproject.finddoc("Extension.Install").getscript
theuninstallscript=theproject.finddoc("Extension.Uninstall").getscript

'make sure install and unistall script are compiled
if ((theinstallscript = nil ) or ( theuninstallscript = nil )) then
  Msgbox.info("Install or Uninstall script are not compiled","")
  return nil
end

'make the extension, extension files typically have a .avx suffix
myextension = extension.make("c:\temp\qual.avx".asfilename, "Water quality",
theinstallscript, theuninstallscript, {})


'Add the view object called "my view"
'myextension.add(theproject.finddoc("Test"))


'Get the new button items and add them to extension

theviewbuttonbar = theproject.findGui("View").getbuttonbar

thebutton1 = theviewbuttonbar.findbyscript("Qual.Flow")
thebutton2 = theviewbuttonbar.findbyscript("Qual.Load")
thebutton3 = theviewbuttonbar.findbyscript("Qual.BMPload")
thebutton4 = theviewbuttonbar.findbyscript("Qual.BMPeff")
thebutton6 = theviewbuttonbar.findbyscript("Qual.BMPfut")
thebutton7 = theviewbuttonbar.findbyscript("Qual.Average")
thebutton8 = theviewbuttonbar.findbyscript("Qual.Zonalmean")
thebutton9 = theviewbuttonbar.findbyscript("Qual.Wshd")
thebutton10 = theviewbuttonbar.findbyscript("Qual.Creek")
thebutton11 = theviewbuttonbar.findbyscript("Qual.Creeklimit")
```

```
thebutton12 = theviewbuttonbar.findbyscript("Qual.Pick")
thebutton13 = theviewbuttonbar.findbyscript("Qual.Mergetheme")
thebutton14 = theviewbuttonbar.findbyscript("Qual.HydroZdlsv")

thetablebuttonbar = theproject.findGui("Table").getbuttonbar

thebutton15 = thetablebuttonbar.findbyscript("Qual.Delete")
thebutton16 = thetablebuttonbar.findbyscript("Qual.Join")


myextension.add(thebutton1)
myextension.add(thebutton2)
myextension.add(thebutton3)
myextension.add(thebutton4)
myextension.add(thebutton6)
myextension.add(thebutton7)
myextension.add(thebutton8)
myextension.add(thebutton9)
myextension.add(thebutton10)
myextension.add(thebutton11)
myextension.add(thebutton12)
myextension.add(thebutton13)
myextension.add(thebutton14)
myextension.add(thebutton15)
myextension.add(thebutton16)


'Get the new tool item and add it to the extension
theviewtoolbar = theproject.findgui("View").gettoolbar
thetool = theviewtoolbar.findbyscript("Qual.Addpoint")
myextension.add(thetool)

'Get the new menu items and add them to the extension

theviewmenubar = theproject.findGui("View").getmenubar
themenu1 = theviewmenubar.findbylabel("Qual")
themenu2 = theviewmenubar.findbylabel("QualTools")

thetablemenubar = theproject.findGui("Table").getmenubar
themenu3 = thetablemenubar.findbylabel("QualTable")


' Make it the last items

myextension.add(themenu1)
myextension.add(themenu2)
myextension.add(themenu3)


'add necessary scripts to the extension, assumes all scripts
'are prefixed with "Qual."

for each d in av.getproject.getdocs
  if (d.is(sed) and (d.getname.left("qual".count+1) = "Qual.")) then
   myextension.add(d.getscript)
  end
end

'Add a description to appear in the extension dialog
myextension.setabout("Water Quality extension")

'Commit change to the extension
myextension.commit
```

'Script: Extension.Uninstall

```
'----------------------------
'--- Creation information ---
'----------------------------

'Name: Extension.uninstall
'Version: 1.0
'Creation date: 11/17/97
'Author: Christine Dartiguenave
'Center for Research in Water Resources
'The University of Texas at Austin
'darti@crwr.utexas.edu


'---------------------------
'--- Purpose/Description ---
'---------------------------

'Installs the water quality extension.


'The self object for this script is the extension
'Check if there is an active project

  if (av.getproject = nil) then
    return nil
  end

'No need to uninstall if project is closing

  if (av.getproject.isclosing) then
    return nil
  end
```

```
'Remove the buttons

'View

thebuttonbar=av.getproject.findGUI("View").getbuttonbar

thebuttonbar.remove(self.get(0))
thebuttonbar.remove(self.get(1))
thebuttonbar.remove(self.get(2))
thebuttonbar.remove(self.get(3))
thebuttonbar.remove(self.get(4))
thebuttonbar.remove(self.get(5))
thebuttonbar.remove(self.get(6))
thebuttonbar.remove(self.get(7))
thebuttonbar.remove(self.get(8))
thebuttonbar.remove(self.get(9))
thebuttonbar.remove(self.get(10))
thebuttonbar.remove(self.get(11))
thebuttonbar.remove(self.get(12))


'Table

thebuttonbar=av.getproject.findGUI("Table").getbuttonbar

thebuttonbar.remove(self.get(13))
thebuttonbar.remove(self.get(14))

'Remove the tool
thetoolbar=av.getproject.findgui("view").gettoolbar
thetoolbar.remove(self.get(15))

'Remove the menus
themenubar=av.getproject.findgui("View").getmenubar
themenubar.remove(self.get(16))
themenubar.remove(self.get(17))

themenubar2=av.getproject.findgui("Table").getmenubar
themenubar2.remove(self.get(18))
```

'Script: Qual.Addpoint

```
'
'
'---------------------------
'--- Creation information ---
'---------------------------
'
'Name: Qual.Addpoint
'Version: 1.0
'Date: 5/20/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'

'--------------------------
'--- Purpose/Description ---
'--------------------------
'
'Create a point coverage in Arcview or add new points to a point coverage.
'The snap command enables to snap the point to a line coverage.

'-------------------
'--- Get the View ---
'-------------------

theview=av.getactivedoc

'----------------------
'--- Enter the point ---
'----------------------
'

ThePntV=theView.GetDisplay.ReturnUserPoint


'-------------------------------------------------
'--- Add the point to an existing point coverage ---
'-------------------------------------------------
'

rel = msgbox.yesno("Do you want to create a new point coverage?" , Script.The.GetName
,true)

 if (rel.not) then
   if (theView.GetThemes.Count = 0) then
     msgbox.error("No active themes found", Script.The.GetName)
     exit
   end

   stList=list.Make
     for each thm in TheView.GetThemes
       if(thm.is(Ftheme))then
         if (thm.GetFtab.GetShapeClass.GetClassName="Point")then
           stList.add(thm)
         end
       end
     end
```

```
    stthm=Msgbox.ChoiceAsString(stList,"Choose a point coverage.",Script.The.GetName)
     if(stthm=nil)then
       exit
     end

    else

'-------------------------------------
'--- Creating a new point coverage ---
'-------------------------------------

outFName = av.GetProject.MakeFileName("Point", "shp")

thefname = FileDialog.Put(outFName, "*.shp", "New Point Coverage")

if (thefName = Nil) then

  exit

end



         theFtab=Ftab.MakeNew(theFName,point)
         PntID=Field.Make("Id",#FIELD_Short,4,0)
         theftab.seteditable(true)
         theFtab.AddFields({PntID})
         stthm=Ftheme.Make(theFtab)
         TheView.AddTheme(stthm)
         stthm.setvisible(true)
       end



 que1 = msgbox.yesno("Snap point to a line coverage?" , Script.The.GetName , true)


'------------
'--- Snap ---
'------------

    if (que1) then


       lList=list.Make
         for each thm in TheView.GetThemes
           if(thm.is(Ftheme))then
             if(thm.GetFtab.GetShapeClass.GetClassName="Polyline")then
               lList.add(thm)
             end
           end
         end


       lthm=Msgbox.ChoiceAsString(lList,"Choose a line coverage.",Script.The.GetName)
       if(lthm=nil)then
         exit
       else
        lthmName=lthm.GetName
       end


     lftab=lThm.GetFtab
     lshp=lftab.FindField("Shape")
     recs=lthm.FindByPoint(ThePntV)
        if(recs.IsEmpty.not)then
          rec=recs.get(0)
          TheShpV=lftab.ReturnValue(lShp,rec)
          theline=theshpv.aspolyline
          m=theline.querypointdistance(thepntv,1000)
          newline=theline.returndensified(10000)
          multi=newline.asmultipoint
          val=thepntv.snap(multi,1000)
            if (val) then
              msgbox.info("Snapped",Script.The.GetName)
            end
         else
            msgbox.info("No line theme found",Script.The.GetName)
            exit
         end
end

theftab = stthm.getftab
theftab.seteditable(true)
therecno = theftab.addrecord
theshapef = theftab.findfield("Shape")
theftab.setvalue(theshapef, therecno, thepntv)
theftab.seteditable(false)


'------------------------------------------
'--- Enter the parameters for the point ---
'------------------------------------------

fieldlist=theftab.getfields
n=theftab.GetNumRecords
m=fieldlist.count
  p=m-1
```

```
 i=0

 minnum=10.min(p)
 newfieldlist=list.make

 for each i in 1..minnum
   thefield=fieldlist.get(i).asstring
   newfieldlist.add(thefield)
   i=i+1
 end

   q=minnum-1
   defaultlist=list.make

 if (newfieldlist.get(0)="id") then

defaultlist={n.asstring}
   if (q > 1) then
     for each i in 1..q
       defaultlist.add("-")
       i=i+1
     end
   end
 else
    if (p>0) then
      for each i in 1..minnum
        defaultlist.add("-")
        i=i+1
      end
    end
 end


   param = MsgBox.MultiInput( stthm.asstring++"fields", Script.The.GetName, newfieldlist,
defaultlist)
   if (param=nil) then
     exit
   end

theftab.seteditable(true)

i=1
j=1
 for each i in 1..n
   for each j in 1..minnum
      ptfield=fieldlist.get(j)
      theftab.setvalue(ptfield,therecno,param.get(j-1))
      j=j+1
   end
   i=i+1
  end
theftab.seteditable(false)


msgbox.info("End",Script.The.GetName)

'-----------
'--- End ---
'-----------
```

' Script Qual.Average

```
'
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.Average
'Version: 1.0
'Date: 6/26/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'---------------------------
'--- Purpose/Description ---
'---------------------------
'
'This program computes the average value at any cell, based on its drainage area.

'-------------------
'--- Get the View ---
'-------------------
'

theView=av.GetActiveDoc

'---------------------
'--- Get the themes ---
'---------------------
'

if (theView.GetThemes.Count = 0) then
   msgbox.error("No themes found", "IC")
   exit
end
```

```
'Choose a theme (grid or polygon coverage)

ThemeList=list.Make
for each thm in TheView.GetThemes
  if(thm.is(Ftheme))then
     if(thm.GetFtab.GetShapeClass.GetClassName="Polygon")then
       ThemeList.add(thm)
     end
  end
  if(thm.is(Gtheme)) then
    ThemeList.add(thm)
   end
end


thethm=Msgbox.ChoiceAsString(ThemeList,"Choose a value theme.",Script.The.GetName)
    if(thethm=nil)then
      exit
    end


'Flow direction grid

gridList=list.Make
 for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
       gridList.add(thm)
    end
 end


fdrthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction
grid.",Script.The.GetName)
    if(fdrthm=nil)then
      exit
    end

fdirgrid=fdrthm.getgrid


 'Flow accumulation grid

 facthm=Msgbox.ChoiceAsString(gridList,"Choose a flow accumulation
grid.",Script.The.GetName)
    if(facthm=nil)then
      exit
    end

facgrid=facthm.getgrid

'Name the new grid

outFname = av.GetProject.MakeFileName("grid", "")
aname = FileDialog.Put(outFName, "", Script.The.GetName)

if (aname = Nil) then
  exit
end

'--------------------------------------
'--- Alternative (grid or coverage) ---
'--------------------------------------

if (thethm.is(Ftheme))then


'---------------------
'--- Get the table ---
'---------------------
'

theFtab=thethm.getFtab
fieldlist=theftab.getfields
thefield = MsgBox.choiceAsString(fieldlist, "Choose a value field" , Script.The.GetName)
if (thefield=nil) then
  MsgBox.Info("No field selected", Script.The.GetName)
  exit
end

'
'-----------------------
'--- Convert to grid ---
'-----------------------

theFtab.seteditable(true)
cellsize=fdirgrid.getcellsize
box=fdirgrid.getextent
aPrj = theView.GetProjection
thegrid = Grid.MakeFromFTab(theFTab, aPrj, theField, {cellSize, box})


 else
   thegrid=thethm.getgrid
 end

'-------------------------
'--- Flow accumulation ---
'-------------------------
```

```
thefacgrid = (fdirgrid.flowaccumulation(thegrid))
theavg1grid=thefacgrid/facgrid
thelist1=list.make
thelist1.add(thegrid)
theavggrid=theavg1grid.merge(thelist1)


theavggrid.savedataset(aname)
theavgthm = gtheme.make(theavggrid)
theview.addtheme(theavgthm)
theavgthm.setlegendvisible(false)


'final message to user

message = "Average grid created."
msgbox.info(message,Script.The.GetName)


'-----------
'--- End ---
'-----------
```

'Script Qual.BMPeff

```
'
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.BMPeff
'Version: 1.0
'Date: 5/21/97
'Modified: 10/16/97
'Author: Christine Dartiguenave
'        Center for Research in Water Resources
'        The University of Texas at Austin
'        darti@crwr.utexas.edu
'
'---------------------------
'--- Purpose/Description ---
'---------------------------
'
'This program computes the load after the action of
'located BMPs defined by their efficiency.

'--------------------
'--- Get the View ---
'--------------------
'

theView=av.GetActiveDoc

'Check if there are any theme in the view.

if (theView.GetThemes.Count = 0) then
  msgbox.error("No themes found", Script.The.GetName)
  exit
end


'---------------------------
'--- Set analysis extent ---
'---------------------------

'bring up the AnalysisPropertiesDialog
theAE = AnalysisPropertiesDialog.Show(theView,FALSE,"Analysis Properties")
  if (theAE=nil) then
    exit
  end

theExtent = Rect.Make(0@0,1@1)
theCellSize = 1
if ((theAE.GetExtent(theExtent) <> #ANALYSISENV_VALUE) or
    (theAE.GetCellSize(theCellSize) <> #ANALYSISENV_VALUE)) then
  theCE = AnalysisPropertiesDialog.Show(theView,TRUE,"Analysis Extent")
   'check for Cancel from dialog
  if (theCE = NIL) then

    return NIL
  end
  theCE.GetCellSize(theCellSize)
  theCE.GetExtent(theExtent)
end

Grid.SetAnalysisCellSize ( #GRID_ENVTYPE_VALUE , theCellSize )
Grid.SetAnalysisExtent ( #GRID_ENVTYPE_VALUE , theextent )



'-----------------------------
'--- Set working directory ---
'-----------------------------

aProject=av.GetProject
```

```
defaultdir=aProject.GetWorkDir
inputdir=MsgBox.Input("Choose the working
directory.",Script.The.GetName,defaultdir.asstring)
  if (inputdir=nil) then
  else
    aDirName = inputdir.asfilename
    aProject.SetWorkDir (aDirName)
  end

'------------------
'--- Get themes ---
'------------------


'Name the new grid

outFName = av.GetProject.MakeFileName("newload", "")

aname = FileDialog.Put(outFName, "", "New load grid")

if (aname = Nil) then

  exit

end


'List the available grids

ThmList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
      ThmList.add(thm)
    end
  end



'Choose a flow direction grid

    fdirthm=Msgbox.ChoiceAsString(ThmList,"Choose a flow direction
grid",Script.The.GetName)
      if(fdirthm=nil)then
        exit
      end

fdirgrid=fdirthm.getgrid


'Choose a flow accumulation grid

  facthm=Msgbox.ChoiceAsString(ThmList,"Choose a flow accumulation
grid",Script.The.GetName)
    if(facthm=nil)then
      exit
    end

  facgrid=facthm.getgrid



'List the available point coverages

Thm1List=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if(thm.GetFtab.GetShapeClass.GetClassName="Point")then
        Thm1List.add(thm)
      end
    end
  end


'Choose the BMPs point coverage

Ptthm=Msgbox.ChoiceAsString(Thm1List,"Choose a BMPs point
coverage",Script.The.GetName)
  if(Ptthm=nil)then

    exit
  end


'Get the point attribute table

pttab = Ptthm.getftab
  if (pttab = nil) then
    msgbox.error("Can't open point theme",Script.The.GetName)
    exit
  end


'Choose the removal efficiency field

ptfieldlist=pttab.getfields

pteff=Msgbox.ChoiceAsString(ptfieldlist,"Choose a BMPs efficiency
field",Script.The.GetName)
if (pteff=nil) then
```

```
  msgbox.error("Can't find efficiency field in point theme",Script.The.GetName)
  exit
end

ptshape = pttab.findfield("shape")
  if (ptshape = nil) then
    msgbox.error("Can't find 'shape' field in point theme",Script.The.GetName)
    exit
  end

'Correct the efficiency field

question=MsgBox.YesNo("Do you want to correct the efficiency field?",
Script.The.GetName , true )
  if (question) then
     bmparea=Msgbox.ChoiceAsString(ptfieldlist,"Choose an observed drainage area field (in
acres).",Script.The.GetName)
    if (bmparea=nil) then
      msgbox.error("Can't find drainage area field in point theme",Script.The.GetName)
    exit
    end
  end



'Choose the initial load grid
loadthm=Msgbox.ChoiceAsString(thmList,"Choose an initial load grid",Script.The.GetName)
  if(loadthm=nil)then
    exit
  end
load0grid = loadthm.getgrid


'Choose a watershed grid

  wshdthm=Msgbox.ChoiceAsString(ThmList,"Choose a watershed zones
grid",Script.The.GetName)
    if(wshdthm=nil)then
      exit
    end

wshdgrid=wshdthm.getgrid



'Create initial load field

ptload0 =  pttab.findfield("load0")
  if (ptload0=nil) then
    pttab.seteditable(true)
    ptload0 = field.make("load0", #FIELD_DECIMAL, 16, 3)
  else
    pttab.seteditable(true)
    ptload0=field.make("load0xxx",#FIELD_DECIMAL, 16, 3)
  end

pttab.addfields({ptload0})
pttab.seteditable(false)



'Write the initial load to the field

pttab.seteditable(true)
  for each rec in pttab
    shapev = pttab.returnvalue(ptshape,rec)
    val = load0grid.cellvalue(shapeV,Prj.MakeNull)
    pttab.setvalue(ptload0,rec,val)
  end
pttab.seteditable(false)


'-----------------------------
'--- Efficiency correction ---
'-----------------------------


if (question) then

'Create a fac field to store the flow accumulation

  ptfac = pttab.findfield("fac")
    if (ptfac = nil) then
      pttab.seteditable(true)
      ptfac = field.make("fac", #FIELD_DECIMAL, 16, 0)
      pttab.addfields({ptfac})
      pttab.seteditable(false)
    end



'Pick flowaccumulation for all points


  pttab.seteditable(true)
    for each rec in pttab
      shapev = pttab.returnvalue(ptshape,rec)
      facval = facgrid.cellvalue(shapeV,Prj.MakeNull)
      pttab.setvalue(ptfac,rec,facval)
    end
  pttab.seteditable(false)
```

```
'Compute computed drainage area (in acre) from the fac value for a 100 ft cellsize

 bmpeff = pttab.findfield("effcorr")
  if (bmpeff = nil ) then
    pttab.seteditable(true)
    bmpeff = field.make("effcorr", #FIELD_DECIMAL, 5, 3)
    pttab.addfields({bmpeff})
    pttab.seteditable(false)
  end

 pttab.seteditable(true)
  for each rec in pttab
    facval = pttab.returnvalue(ptfac,rec)
    areaobsval = pttab.returnvalue(bmparea,rec)
    eff = pttab.returnvalue(pteff,rec)
    areacompval = facval * 10000 / 43650
    corrarea= areaobsval / areacompval
    effcorr = corrarea * eff
    pttab.setvalue(bmpeff, rec,effcorr)
  end
 pttab.seteditable(false)


else
bmpeff=pteff
end

'-----------------------------------------------------------
'--- Write the new load at the BMP's to the field removed ---
'-----------------------------------------------------------
'


'Check the existence of the field where the removed and the remaining loads will be writen to

ptremoved = pttab.findfield("removed")
 if (ptremoved = nil) then
   pttab.seteditable(true)
   ptremoved = field.make("removed", #FIELD_DECIMAL, 16, 3)
   pttab.addfields({ptremoved})
   pttab.seteditable(false)
 else
 end


ptremain = pttab.findfield("remain")
 if (ptremain = nil) then
   pttab.seteditable(true)
   ptremain = field.make("remain", #FIELD_DECIMAL, 16, 3)
   pttab.addfields({ptremain})
   pttab.seteditable(false)
 else
 end


'Check the efficiency: percentage or decimal fraction

      effmax=0
      effmin=0
       for each rec in pttab
         effval = pttab.returnvalue(pteff,rec)
         effmax = effmax.max(effval)
         effmin = effmin.min(effval)
       end
       if (effmax > 1) then
         percval = 100
       elseif (effmin < -1) then
         percval = 100
       else
         percval = 1
       end


'Compute the removed and remaining loads and write them in the table


pttab.seteditable(true)
 for each rec in pttab
   bmpval=pttab.returnvalue(bmpeff,rec)
   load0val=pttab.returnvalue(ptload0,rec)
   removedloadval = load0val*bmpval/percval
   remainloadval = load0val - removedloadval
   pttab.setvalue(ptremoved,rec,removedloadval)
   pttab.setvalue(ptremain,rec,remainloadval)
 end
pttab.seteditable(false)



'---------------------------------------
'--- Check if the stations are nested ---
'---------------------------------------


'Compute the flow accumulation: removed load value at the BMP's, 0 everywhere else
'if the flowaccumulation value is 0 at the BMP's, non nested
```

```
'Create grids with the removed and remaining load value at the BMP's
aPrj = theView.GetProjection
ptremoved = pttab.findfield("removed")
removedgrid = Grid.MakeFromFTab(pttab, aPrj, ptremoved, {thecellSize, theextent})
ptremain = pttab.findfield("remain")
remaingrid = Grid.MakeFromFTab(pttab, aPrj, ptremain, {thecellSize, theextent})


'Compute the weighted flowaccumulation

rmloadgrid = (fdirgrid.flowaccumulation(removedgrid))

'Check the existence of a field for the weighted flowaccumulation

ptrmfac = pttab.findfield("rmfac")
  if (ptrmfac = nil) then
    pttab.seteditable(true)
    ptrmfac = field.make("rmfac", #FIELD_DECIMAL, 16, 0)
    pttab.addfields({ptrmfac})
    pttab.seteditable(false)
  else
  end


'Pick the flow accumulation value at the BMP's

n=0
pttab.seteditable(true)
for each rec in pttab
  shapev = pttab.returnvalue(ptshape,rec)
  val = rmloadgrid.cellvalue(shapeV,Prj.MakeNull)
  pttab.setvalue(ptrmfac,rec,val)
  n=n.max(val)
end
pttab.seteditable(false)


'Check if there are some nested BMP's and compute the new load

if (n=0) then 'no nested BMP's
  tmploadgrid= (load0grid - rmloadgrid )
  listgrid=list.make
  listgrid.add(tmploadgrid)
  newloadgrid=remaingrid.merge(listgrid)
  newloadgrid.savedataset(aname)
  newloadthm=Gtheme.Make(newloadgrid)
  theview.addtheme(newloadthm)

'Remove the fields

fieldlist= {ptload0,ptremoved,ptremain,ptrmfac}
  if (question) then
    fieldlist.add(ptfac)
    fieldlist.add(bmpeff)
  end
pttab.seteditable(true)
ptTab.RemoveFields (fieldlist)
pttab.seteditable(false)


  msgbox.info("The new load has been calculated.",Script.The.GetName)
  exit

else 'order the BMP's (0 if non nested, 1, 2 for different flow accumulation runs)



'Create a field wshd based on the watersheds

ptwshd = pttab.findfield("wshd")
  if (ptwshd = nil) then
    pttab.seteditable(true)
    ptwshd = field.make("wshd", #FIELD_DECIMAL, 4, 0)
    pttab.addfields({ptwshd})
    pttab.seteditable(false)
  else
  end



'Take the id-number associated with each watershed

pttab.seteditable(true)
  for each rec in pttab
    shapev = pttab.returnvalue(ptshape,rec)
    val = wshdgrid.cellvalue(shapeV,Prj.MakeNull)
    pttab.setvalue(ptwshd,rec,val)
  end
pttab.seteditable(false)



'Create a new field maxnum for the maximum number of nested BMP's in each watershed
```

```
pttab.seteditable(true)
ptmaxnum=pttab.findfield("maxnum")
  if (ptmaxnum=nil) then
    pttab.seteditable(true)
    ptmaxnum = field.make("maxnum", #FIELD_DECIMAL, 4, 0)
    pttab.addfields({ptmaxnum})
  else
  end

'for each rec in pttab
 ' pttab.setvalue(ptmaxnum, rec, 0)
'end



'Select the watersheds in which there are nested BMP's

wshdlist=list.make

for each rec in pttab
  rmfacval=pttab.returnvalue(ptrmfac,rec)
    if (rmfacval > 0) then 'it is a nested BMP
      wshdval=pttab.returnvalue(ptwshd,rec)
      wshdlist.add(wshdval)
    end
end


'Assign the number of BMP's per watersheds in a new field maxnum (0 for the normal ones)

maxnumber=0
for each rec in pttab
  rmfacval=pttab.returnvalue(ptrmfac,rec)
    if (rmfacval > 0) then 'it is a nested BMP
      wshdval=pttab.returnvalue(ptwshd,rec)
      i=0
        for each rec in wshdlist
          if (wshdval=rec) then
            i=i+1
          else
          end
        end
      pttab.setvalue(ptmaxnum, rec, i)
      maxnumber=maxnumber.max(i)
    end
end
'msgbox.info("max= "++maxnumber.asstring,"")

'If the maximum number of nested BMP's in any watershed is 1

if (maxnumber=1) then

'Create two new efficiency fields (2 runs)

  ptneweff0 =  pttab.findfield("neweff0")
    if (ptneweff0=nil) then
      pttab.seteditable(true)
      ptneweff0 = field.make("neweff0", #FIELD_DECIMAL, 3, 2)
      pttab.addfields({ptneweff0})
      pttab.seteditable(false)
    end



  ptneweff1 =  pttab.findfield("neweff1")
    if (ptneweff1=nil) then
      pttab.seteditable(true)
      ptneweff1 = field.make("neweff1", #FIELD_DECIMAL, 3, 2)
      pttab.addfields({ptneweff1})
      pttab.seteditable(false)
    end



'Write the efficiency values

  pttab.seteditable(true)
    for each rec in pttab
      effval=pttab.returnvalue(bmpeff,rec)
      rmfacval=pttab.returnvalue(ptrmfac,rec)
        if (rmfacval > 0) then'nested BMPs
          pttab.setvalue(ptneweff1,rec,effval)
        else
          pttab.setvalue(ptneweff0,rec,effval)
        end
    end



'Compute the new load

  load1grid=av.run("Qual.bmpeff_cpt",{load0grid,fdirgrid,pttab,ptneweff0,percval})
  newloadgrid=av.run("Qual.bmpeff_cpt",{load1grid,fdirgrid,pttab,ptneweff1,percval})

  newloadgrid.savedataset(aname)
  newloadthm=Gtheme.Make(newloadgrid)
  theView.AddTheme(newloadthm)
  newloadthm.setlegendvisible(false)
```

```
'Remove the fields

fieldlist= {ptload0,ptremoved,ptremain,ptmaxnum,ptneweff0,ptneweff1,ptrmfac,ptwshd}
 if (question) then
   fieldlist.add(ptfac)
   fieldlist.add(bmpeff)
 end
pttab.seteditable(true)
ptTab.RemoveFields (fieldlist)
pttab.seteditable(false)



 msgbox.info("The new load has been calculated.","")
 exit


else 'if there are more than 1 nested BMP in each watershed, sort by flowacumulation.



if (question=false) then


'Create a fac field to store the flow accumulation

 ptfac = pttab.findfield("fac")
   if (ptfac = nil) then
     pttab.seteditable(true)
     ptfac = field.make("fac", #FIELD_DECIMAL, 16, 0)
     pttab.addfields({ptfac})
     pttab.seteditable(false)
   end



'Pick flowaccumulation for all points


 pttab.seteditable(true)
   for each rec in pttab
     shapev = pttab.returnvalue(ptshape,rec)
     facval = facgrid.cellvalue(shapeV,Prj.MakeNull)
     pttab.setvalue(ptfac,rec,facval)
   end
 pttab.seteditable(false)

end
```

```
'Check the existence of a field order to sort the BMP's

 ptorder=pttab.findfield("order")
   if (ptorder = nil) then
     pttab.seteditable(true)
     ptorder = field.make("order", #FIELD_DECIMAL, 4, 0)
     pttab.addfields({ptorder})
     pttab.seteditable(false)
   end

'Create list of BMP's in the same watershed


 minimum=1000000000
 wshdlist.removeduplicates
 idlist=list.make

   for each wshd in wshdlist
     i=0
     for each rec in pttab
       rmfacval=pttab.returnvalue(ptrmfac,rec)
         if (rmfacval > 0) then 'it is a nested bmp
           wshdval=pttab.returnvalue(ptwshd,rec)
             if (wshdval=wshd) then
               idlist.add(i)
             end
         else
           pttab.seteditable(true)
           pttab.setvalue(ptorder,rec,0)
         end
       i=i+1
      end


'Compute the minimum for the list


   faclist=list.make
     for each rec in idlist
       facval=pttab.returnvalue(ptfac,rec)
       faclist.add(facval)
       'msgbox.info("facval"+facval.asstring,"")
       minimum=minimum.min(facval)
     end
   'msgbox.info("the min1 is "++minimum.asstring,"")
```

```
'Order the nested bmp in a selected watershed according to their
'flowaccumulation value


  pttab.seteditable(true)
  n=idlist.count
  i=1


    for each i in 1..n
      idlist1=list.make
        for each rec in idlist
          facval=pttab.returnvalue(ptfac,rec)
            if(facval=minimum) then
              pttab.seteditable(true)
              pttab.setvalue(ptorder,rec,i)
              pttab.seteditable(false)
            else
              idlist1.add(rec)
            end
        end
      i=i+1
      minimum=100000000
        for each rec in idlist1
          facval=pttab.returnvalue(ptfac,rec)
          minimum=minimum.min(facval)
        end
      idlist=idlist1
    end
 end



'Create as many neweff fields as necessary

 i=0
  for each i in 0..maxnumber
    ptneweff =  pttab.findfield("neweff")
      if (ptneweff=nil) then
        pttab.seteditable(true)
        ptneweff = field.make("neweff", #FIELD_DECIMAL, 3, 2)
        pttab.addfields({ptneweff})
        pttab.seteditable(false)
      end

    pttab.seteditable(true)
      for each rec in pttab
        orderval=pttab.returnvalue(ptorder,rec)
            if (orderval=i) then
              bmpeffval=pttab.returnvalue(bmpeff,rec)
              pttab.setvalue(ptneweff,rec,bmpeffval)
            else
              pttab.setvalue(ptneweff,rec,0)
            end
          end


'Compute the new loads


      load0grid=av.run("Qual.bmpeff_cpt",{load0grid,fdirgrid,pttab,ptneweff,percval})
      i=i+1
    end

'Save the load grid

  load0grid.savedataset(aname)
  newloadthm=Gtheme.Make(load0grid)
  theview.addtheme(newloadthm)
  newloadthm.setlegendvisible(false)

'Remove the fields

fieldlist= {ptload0,ptremoved,ptremain,ptmaxnum,ptfac,ptneweff,ptrmfac,ptwshd,ptorder}
  if (question) then
    fieldlist.add(ptfac)
    fieldlist.add(bmpeff)
  end
pttab.seteditable(true)
ptTab.RemoveFields (fieldlist)
pttab.seteditable(false)


  msgbox.info("The new load has been calculated.","")
  exit

end
end



'-----------
'--- End ---
'-----------
```

'Script Qual.BMPeff_cpt

```
'
'---------------------------
'--- Creation information ---
'---------------------------
'
'Name: Qual.BMPeff_cpt
'Version: 1.0
'Date: 05/23/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'--------------------------
'--- Purpose/Description ---
'--------------------------
'
'This program takes a load grid and a point coverage
'and computes the new load for non nested BMP's.
'It is used as a subroutine of Qual.BMPeff.



theview=av.getactivedoc


if (self.count=5) then
  load0grid=self.get(0)
  fdirgrid=self.get(1)
  pttab=self.get(2)
  bmpeff=self.get(3)
  percval=self.get(4)

  else
exit
end


ptshape = pttab.findfield("shape")
  if (ptshape = nil) then
    msgbox.error("Can't find 'shape' field in point theme",Script.The.GetName)
    exit
  end

'-----------------------------------------------
'--- Write the initial load to the field load0 ---
'-----------------------------------------------


'Check the existence of the initial load field load0

ptload0 =  pttab.findfield("load0")
if (ptload0=nil) then
  pttab.seteditable(true)
  ptload0 = field.make("load0", #FIELD_DECIMAL, 16, 0)
  pttab.addfields({ptload0})
  pttab.seteditable(false)
end


'Write the initial load to the field load0


pttab.seteditable(true)
  for each rec in pttab
    shapev = pttab.returnvalue(ptshape,rec)
    val = load0grid.cellvalue(shapeV,Prj.MakeNull)
    pttab.setvalue(ptload0,rec,val)
  end
pttab.seteditable(false)


'---------------------------------------------------------
'--- Write the new load at the BMP's to the field removed ---
'---------------------------------------------------------
'


ptremoved=pttab.findfield("removed")
ptremain=pttab.findfield("remain")

'Compute the removed and remaining loads and write them in the table


pttab.seteditable(true)
for each rec in pttab
    bmpval=pttab.returnvalue(bmpeff,rec)
    load0val=pttab.returnvalue(ptload0,rec)
    removedloadval = load0val*bmpval/percval
    remainloadval = load0val-removedloadval
    pttab.setvalue(ptremoved,rec,removedloadval)
    pttab.setvalue(ptremain,rec,remainloadval)
    if (bmpval = 0) then
        pttab.setvalue(ptremain,rec,0)
```

```
   end
end

pttab.seteditable(false)

'Create a grid with the remaining value just at the station with non negative neweff
'Create grids with the removed and remaining load value at the BMP's

cellsize=load0grid.getcellsize
box=load0grid.getextent
aPrj = theView.GetProjection
ptremain=pttab.findfield("remain")
remain1grid = Grid.MakeFromFTab(pttab, aPrj, ptremain, {cellSize, box})
remaingrid=(remain1grid = 0.asgrid).setnull(remain1grid)
ptremoved = pttab.findfield("removed")
removedgrid = Grid.MakeFromFTab(pttab, aPrj, ptremoved, {cellSize, box})


'Compute the weighted flowaccumulation

rmloadgrid = (fdirgrid.flowaccumulation(removedgrid))

'Compute the new load

tmploadgrid=load0grid-rmloadgrid
listgrid=list.make
listgrid.add(tmploadgrid)
newloadgrid = remaingrid.merge( listgrid )



return newloadgrid
```

'Script: Qual.BMPfut

```
'
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.BMPfut
'Version: 1.0
'Date: 10/17/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'---------------------------
'--- Purpose/Description ---
'---------------------------
'
'This program compute the new load after implementation of the future BMPs (areal
representation).

'--------------------
'--- Get the View ---
'--------------------
'


theView=av.GetActiveDoc
aPrj = theView.GetProjection

'Check if there are any theme in the view.

  if (theView.GetThemes.Count = 0) then
    msgbox.error("No themes found", Script.The.GetName)
    exit
  end

'---------------------------
'--- Set analysis extent ---
'---------------------------

'bring up the AnalysisPropertiesDialog
theAE = AnalysisPropertiesDialog.Show(theView,FALSE,"Analysis Properties")
  if (theAE=nil) then
    exit
  end

theExtent = Rect.Make(0@0,1@1)
theCellSize = 1
if ((theAE.GetExtent(theExtent) <> #ANALYSISENV_VALUE) or
    (theAE.GetCellSize(theCellSize) <> #ANALYSISENV_VALUE)) then
  theCE = AnalysisPropertiesDialog.Show(theView,TRUE,"Analysis Extent")
  'check for Cancel from dialog
  if (theCE = NIL) then

    return NIL
  end
  theCE.GetCellSize(theCellSize)
  theCE.GetExtent(theExtent)
end

Grid.SetAnalysisCellSize ( #GRID_ENVTYPE_VALUE , theCellSize )
Grid.SetAnalysisExtent ( #GRID_ENVTYPE_VALUE , theextent )
```

```
'-----------------------------
'--- Set working directory ---
'-----------------------------

aProject=av.GetProject
defaultdir=aProject.GetWorkDir
inputdir=MsgBox.Input("Choose the working
directory.",Script.The.GetName,defaultdir.asstring)
  if (inputdir=nil) then
  else
    aDirName = inputdir.asfilename
    aProject.SetWorkDir (aDirName)
  end


recharge=msgbox.yesno("Do you want to consider a recharge
zone",Script.The.GetName,true)


'----------------------------------------
'--- Choose the constituents to model ---
'----------------------------------------

'----------------------
'--- Get the tables ---
'----------------------

doculist = av.GetProject.Getdocs
tablelist=List.Make
  for each d in doculist
    if(d.Is(Table))then
      TableList.add(d)
    end
  end


'Annual capture volume tables

acvtable=Msgbox.ChoiceAsString(tableList,"Choose a capture volume
table.",Script.The.GetName)
  if(acvtable=nil)then
    msgbox.error("No ACV table selected",Script.The.GetName)
    exit
  end
acvtab=acvtable.getvtab
acvlist=acvtab.getfields


'BMPs zones tables

bmptable=Msgbox.ChoiceAsString(tableList,"Choose a BMPs zones
table.",Script.The.GetName)
  if(bmptable=nil)then
    msgbox.error("No BMPs zones table selected",Script.The.GetName)
    exit
  end
bmptab=bmptable.getvtab
bmplist=bmptab.getfields



'Efficiency

efftable=Msgbox.ChoiceAsString(tableList,"Choose an efficiency
table.",Script.The.GetName)
  if(efftable=nil)then
    msgbox.error("No efficiency table selected",Script.The.GetName)
    exit
  end
efftab=efftable.getvtab
efflist=efftab.getfields


'Direct runoff EMC table

emcruntable=Msgbox.ChoiceAsString(tableList,"Choose a direct runoff EMCs
table",Script.The.GetName)
    if(emcruntable=nil)then
      msgbox.error("No direct runoff EMCs table selected",Script.The.GetName)
      exit
    end
emcruntab=emcruntable.getvtab
emcrunlist=emcruntab.getfields
runcons=emcrunlist.get(0)



'List the available grids

gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
      gridList.add(thm)
    end
  end
```

```
'Choose the constituents to model
'Number of constituents

i=0
  for each rec in efftab
    i=i+1
  end
r=i
s=r-1

constfield=efflist.get(0)
conslist=list.make

  for each k in 2..s
    constituent=efftab.returnvaluestring(constfield,k)
    conslist.add(constituent)
  end

choices = MsgBox.MultiListAsString( conslist, "Choose the constituent(s) to model",
Script.The.GetName )
  if (choices = nil) then
    msgbox.info("No constituent selected.", Script.The.GetName)
    exit
  else
    namelist=list.make
    loadlist=list.make

      for each cons in choices
        outFName = av.GetProject.MakeFileName(cons, "")
        aName = FileDialog.Put(outFName, "", cons)
          if (aName = Nil) then
            exit
          end
        namelist.add(aname)

        'Choose an original load grid

        loadthm=Msgbox.ChoiceAsString(gridList,"Choose an initial load grid
for"++cons.asstring,Script.The.GetName)
          if(loadthm=nil)then
            exit
          end
        loadlist.add(loadthm)
    end
  end



'Choose an impervious cover theme

icList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if(thm.GetFtab.GetShapeClass.GetClassName="polygon")then
       icList.add(thm)
      end
    else
      if (thm.is(Gtheme)) then
       iclist.add(thm)
      end

   end
  end



icthm=Msgbox.ChoiceAsString(icList,"Choose an impervious cover
theme.",Script.The.GetName)
    if(icthm=nil)then
      exit
    end

if (icthm.is(ftheme)) then
  anftab=icthm.getftab
  fieldlist=anftab.getfields
  icfield=Msgbox.ChoiceAsString(fieldlist,"Choose the ic field.",Script.The.GetName)
  if(icfield=nil)then
    exit
  end
end

'Choose a flow direction grid


fdirthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction
grid.",Script.The.GetName)
  if(fdirthm=nil)then
    exit
  end

fdirgrid=fdirthm.getgrid



'Choose a bmp zone grid

bmpzonethm=Msgbox.ChoiceAsString(gridList,"Choose a BMPs zones
grid.",Script.The.GetName)
  if(bmpzonethm=nil)then
```

```
    exit
  end
bmpzone=bmpzonethm.getgrid



'Choose a corrected cell runoff grid

runcellthm=Msgbox.ChoiceAsString(gridList,"Choose a corrected cell runoff
grid.",Script.The.GetName)
      if(runcellthm=nil)then
        exit
      end

runcell=runcellthm.getgrid


'Choose a buildup grid

buildupthm=Msgbox.ChoiceAsString(gridList,"Choose a buildup grid.",Script.The.GetName)
      if(buildupthm=nil)then
        exit
      end

buildup=buildupthm.getgrid

'Choose a water land use zone

zonethm=Msgbox.ChoiceAsString(gridList,"Choose a water landuse theme
(zone_gr).",Script.The.GetName)
    if(zonethm=nil)then
      exit
    end

zone=zonethm.getgrid




if (recharge) then


'Choose a total flow grid with recharge

tflowthm=Msgbox.ChoiceAsString(gridList,"Choose a predicted flow grid (with recharge,
flow1).",Script.The.GetName)
      if(tflowthm=nil)then
        exit
      end
tflow=tflowthm.getgrid



'Choose a total flow grid without recharge

tflow0thm=Msgbox.ChoiceAsString(gridList,"Choose a total flow grid (without recharge,
tflow01).",Script.The.GetName)
      if(tflow0thm=nil)then
        exit
      end

totalflow0=tflow0thm.getgrid


'Choose a cell correction recharge

lcorrrechthm=Msgbox.ChoiceAsString(gridList,"Choose a cell correction recharge
grid.",Script.The.GetName)
      if(lcorrrechthm=nil)then
        exit
      end

lcorr_rech=lcorrrechthm.getgrid

end


if (icthm.is(gTheme)) then
  icgrid=icthm.getgrid
else
  anftab=icthm.getftab
  icgrid = Grid.MakeFromFTab(anFTab, aPrj, icfield, {thecellSize, theextent})
end


'----------------------
'--- Capture volumes ---
'----------------------


'number of acv
p=acvlist.count
q=p-1


for each i in 0..q
  thefield=acvlist.get(i)
  a=acvtab.returnvalue(thefield,0)
  b=acvtab.returnvalue(thefield,1)
```

```
  acvgrid=a.asgrid*icgrid+b.asgrid
  acvname = av.getproject.makefilename("acv", "")
  acvgrid.savedataset(acvname)
  acvgthm=Gtheme.Make(acvgrid)
  theview.addtheme(acvgthm)
  acvgthm.setlegendvisible(false)
  i=i+1
end


'------------------
'--- Efficiency ---
'------------------


i=0
  for each rec in bmptab
    i=i+1
  end
n=i

'n is the number of bmp zones
'q total number of bmps
p=bmplist.count
q=p-1



'for each constituent
'number of constituents to model
z=choices.count


if (recharge=true) then

'------------------------------
'--- Flow removal efficiency ---
'------------------------------

'Check if there are non discharge BMPs
theval=0
  for each i in 1..q
    thefield2=efflist.get(i)
    effval=efftab.returnvalue(thefield2,1)
    theval=theval.max(effval)
    i=i+1
  end

  if (theval<>0) then
```

```
    floweffgrid=0.asgrid

     'for each zone
     for each i in 1..n
       floweff=0.asgrid
        for each j in 1..q
          'get the percentage of the bmp
          thefield1=bmplist.get(j)
          bmpval=bmptab.returnvalue(thefield1,i-1)
          'msgbox.info(bmpval.asstring,"bmp%")
          'get the efficiency
          thefield2=efflist.get(j)
          effval=efftab.returnvalue(thefield2,1)
          acvval=efftab.returnvalue(thefield2,0)
          theacvname="acv"+acvval.asstring
          'msgbox.info(theacvname, "acv")
          'msgbox.info(effval.asstring,"removal eff")
          acvthm=theview.findtheme(theacvname)
          theacvgrid=acvthm.getgrid
          floweff=floweff+(bmpval.asgrid*effval.asgrid*theacvgrid)
          j=j+1
        end
      floweffgrid=(bmpzone=i.asgrid).con(floweff,floweffgrid)
      i=i+1
    end



  'thename = av.getproject.makefilename("floweff","")
  'floweffgrid.savedataset(thename)
  'floweffg=Gtheme.Make(floweffgrid)
  'theview.addtheme(floweffg)
  'floweffg.setlegendvisible(false)

  'exit


  rmruncell = runcell * floweffgrid * buildup
  rmflow = (fdirgrid.flowaccumulation(rmruncell))
  totalflow1 = totalflow0 - rmflow
  newflow1 = tflow - rmflow

  thename = av.getproject.makefilename("newflow","")
  newflow1.savedataset(thename)
  newflow1gthm=Gtheme.Make(newflow1)
  theview.addtheme(newflow1gthm)
  newflow1gthm.setlegendvisible(false)
```

```
 else
   totalflow1=totalflow0
 end
 end



'------------------------
'--- Direct Runoff EMC ---
'------------------------


for each l in 1..z
 theeffgrid=0.asgrid
 cons=choices.get(l-1)


'Check storm runoff field

if (icthm.is(ftheme)) then

    anftab.seteditable(true)
    consfield = anftab.findfield(cons+"_[mg/l]")
      if (consfield = nil) then
        consfield = field.make(cons+"_[mg/l]", #FIELD_DECIMAL, 6, 3)
        anftab.addfields({consfield})
      end

end

'Get the parameters a and b (emc=a+b*ic,0<ic<1)
      i=0
      for each rec in emcruntab
        runconsname=emcruntab.returnvaluestring(runcons,rec)
          if (runconsname=cons) then
            p=i
          else
            i=i+1
          end
      end

      afield=emcrunlist.get(1)
      bfield=emcrunlist.get(2)
      a=emcruntab.returnvalue(afield,p)
      b=emcruntab.returnvalue(bfield,p)

      if (icthm.is(Ftheme)) then
        icmax=0
        for each rec in anFTab
          ic1 = anFtab.returnvalue(icfield,rec)
          icmax=icmax.max(ic1)
        end
      if (icmax>1)then
       icperc = true
      else
       icperc = false
      end

        for each rec in anFtab
          ic1 = anFtab.returnvalue(icfield, rec)
            if (icperc=true) then
              ic1=ic1/100
            end

          emcrun=b*ic1+a
          anftab.setvalue(consfield , rec , emcrun )
        end
      else
      aprj=theview.getprojection
      icint=icgrid.int
      icvtab=icint.getvtab
      icfield=icvtab.findfield("value")
        icmax=0
        for each rec in icvtab
          icvalue=icvtab.returnvalue(icfield,rec)
          icmax=icmax.max(icvalue)
        end
          if (icmax<=1) then
            emc_gr0 = icgrid*b + a.asgrid
          else
            emc_gr0 = icgrid*b*0.01 + a.asgrid
          end

      end

aPrj=theview.getprojection
  if (icthm.is(Ftheme)) then
    emc_gr0 = Grid.MakeFromFTab(anFTab, aPrj, consfield, {thecellSize, theextent})
  end

'Set the EMC for water to zero.
emc_gr=(zone=999).con(0.asgrid,emc_gr0)
```

```
'for each zone
for each i in 1..n
      eff=0.asgrid
'for each BMP
   for each j in 1..q
    'get the percentage of the bmp
    thefield1=bmplist.get(j)
    bmpval=bmptab.returnvalue(thefield1,i-1)
    'msgbox.info(bmpval.asstring,"bmp%")
    'get the efficiency
    thefield2=efflist.get(j)
    effval=efftab.returnvalue(thefield2,l+1)
    'msgbox.info(effval.asstring,"eff")
    acvval=efftab.returnvalue(thefield2,0)
    theacvname="acv"+acvval.asstring
    'msgbox.info(theacvname,"acv")
    acvthm=theview.findtheme(theacvname)
    theacvgrid=acvthm.getgrid
    '1=bod in that case
     eff=eff+ (theacvgrid*bmpval.asgrid*effval.asgrid)
    j=j+1
  end
  theeffgrid=(bmpzone=i.asgrid).con(eff,theeffgrid)
  i=i+1

end
 ' theeffgrid.savedataset(aname)
 ' effthm=Gtheme.Make(theeffgrid)
 ' theview.addtheme(effthm)
 ' effthm.setlegendvisible(false)



'Check if effgrid contains negative values
effgrid1 = theeffgrid*10000
effgridint = effgrid1.int
  effgridtab=effgridint.getvtab
  effgridfield=effgridtab.findfield("value")
  themax=0
  themin=0
    for each rec in effgridtab
      theval=effgridtab.returnvalue(effgridfield,rec)
      themax=themax.max(theval)
      themin=themin.min(theval)
    end

  if (themin<0) then
    if (themax=0) then
     effgrid = - theeffgrid
     rmloadcell=runcell*emc_gr*effgrid*buildup*3.048*3.048*3.048*365*86400/1000000
     rmload0=-(fdirgrid.flowaccumulation(rmloadcell))

    else
      neggrid=(theeffgrid<0).setnull(-theeffgrid)
      posgrid=(theeffgrid>=0).setnull(theeffgrid)
      negloadcell = runcell
*emc_gr*neggrid*buildup*3.048*3.048*3.048*365*86400/1000000
      posloadcell = runcell *emc_gr*posgrid*buildup*3.048*3.048*3.048*365*86400/1000000
      negload0 = (fdirgrid.flowaccumulation(negloadcell))
      posload0 = (fdirgrid.flowaccumulation(posloadcell))
      rmload0 = posload0 - negload0

   end
  else
   rmloadcell=runcell*emc_gr*theeffgrid*buildup*3.048*3.048*3.048*365*86400/1000000
   rmload0=(fdirgrid.flowaccumulation(rmloadcell))
  end


'----------------------------------
'--- Compute the corrected load ---
'----------------------------------

loadthm=loadlist.get(l-1)
load=loadthm.getgrid

if (recharge=false) then

'Without recharge

  newload=load-rmload0
  aname=namelist.get(l-1)
  newload.savedataset(aname)
  newloadgthm=Gtheme.Make(newload)
  theview.addtheme(newloadgthm)
  newloadgthm.setlegendvisible(false)

else


'With recharge
  co=load/tflow
  load01 = co * totalflow0 - rmload0
  co1 = load01 / totalflow1
  loadrech = lcorr_rech * co1
  rechload=(fdirgrid.flowaccumulation(loadrech))
  newload=load01-rechload
  aname=namelist.get(l-1)
  newload.savedataset(aname)
  newloadgthm=Gtheme.Make(newload)
```

```
  theview.addtheme(newloadgthm)
  newloadgthm.setlegendvisible(false)


end
l=l+1
end

p=acvlist.count
  i=1
  for each i in 1..p
    theacvname="acv"+i.asstring
    thename=theacvname.asfilename
    acvthm=theview.findtheme(theacvname)
    theview.deletetheme(acvthm)
    grid.deletedataset(thename)
    i=i+1
  end


msgbox.info("Corrected load(s) calculated.",Script.The.GetName)
```

'Script Qual.BMPload

```
'
----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.BMPload
'Version: 1.0
'Date: 5/21/97
'Modified: 10/16/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'--------------------------
'--- Purpose/Description ---
'--------------------------
'
'This program computes the newload after removing
'the mass given in a bmp table corresponding to a bmp coverage.

'-------------------
'--- Get the View ---
'-------------------


theView=av.GetActiveDoc

'Check if there are some themes in the view.

if (theView.GetThemes.Count = 0) then
  msgbox.error("No themes found", Script.The.GetName)
  exit
end


'----------------------------
'--- Set working directory ---
'----------------------------

aProject=av.GetProject
defaultdir=aProject.GetWorkDir
inputdir=MsgBox.Input("Choose the working
directory.",Script.The.GetName,defaultdir.asstring)
  if (inputdir=nil) then
  else
    aDirName = inputdir.asfilename
    aProject.SetWorkDir (aDirName)
  end




'Name the new grid

outFName = av.GetProject.MakeFileName("remload", "")
aname = FileDialog.Put(outFName, "", "Load removed")

if (aname = Nil) then

  exit

end

'-----------------
'--- Get themes ---
'-----------------


'List the available grids
```

```
ThmList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
      ThmList.add(thm)
    end
  end



'Choose a flow direction grid

    fdirthm=Msgbox.ChoiceAsString(ThmList,"Choose a flow direction
grid",Script.The.GetName)
      if(fdirthm=nil)then
        exit
      end

fdirgrid=fdirthm.getgrid

'List the available point coverages

Thm1List=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if(thm.GetFtab.GetShapeClass.GetClassName="Point")then
        Thm1List.add(thm)
      end
    end
  end


'Choose the BMPs point coverage

Ptthm=Msgbox.ChoiceAsString(Thm1List,"Choose a BMPs point
coverage",Script.The.GetName)
  if(Ptthm=nil)then

    exit
  end


'Get the point attribute table

pttab = Ptthm.getftab
  if (pttab = nil) then
    msgbox.error("Can't open point theme",Script.The.GetName)
    exit
  end

'Choose the removed load field

ptfieldlist=pttab.getfields

bmpload=Msgbox.ChoiceAsString(ptfieldlist,"Choose the removed load
field",Script.The.GetName)
if (bmpload=nil) then
  msgbox.error("Can't find removed load field in point theme",Script.The.GetName)
  exit
end

ptshape = pttab.findfield("shape")
  if (ptshape = nil) then
    msgbox.error("Can't find 'shape' field in point theme",Script.The.GetName)
    exit
  end



'Create a grid with the removed value at the BMPs (positive values)

cellsize=fdirgrid.getcellsize
box=fdirgrid.getextent
aPrj = theView.GetProjection
removedgrid = Grid.MakeFromFTab(pttab, aPrj, bmpload, {cellSize, box})

themax=0
themin=0
  for each rec in pttab
    theval=pttab.returnvalue(bmpload,rec)
    themax=themax.max(theval)
    themin=themin.min(theval)
  end

  if (themin<0) then
    if (themax=0) then
      removed1grid=-removedgrid
      rmloadgrid =- (fdirgrid.flowaccumulation(removed1grid))
    else
      neggrid=(removedgrid>=0).setnull(-removedgrid)
      posgrid=(removedgrid<0).setnull(removedgrid)
      negrmload = (fdirgrid.flowaccumulation(neggrid))
      posrmload = (fdirgrid.flowaccumulation(posgrid))
      rmloadgrid=posrmload - negrmload
    end
   else
     rmloadgrid = (fdirgrid.flowaccumulation(removedgrid))
  end

listgrid=list.make
```

```
zerogrid=rmloadgrid*0.asgrid
listgrid.add(zerogrid)
newremgrid = removedgrid.merge(listgrid)
rmgrid= rmloadgrid+newremgrid
rmgrid.savedataset(aname)
rmthm=Gtheme.Make(rmgrid)
theview.addtheme(rmthm)
rmthm.setlegendvisible(false)



Msgbox.info("The load removed by the BMPs has been calculated.",Script.The.GetName)


'-----------
'--- End ---
'-----------
```

'Script: Qual.Creek

```
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.Creek
'Version: 1.0
'Date: 11/17/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'--------------------------
'--- Purpose/Description ---
'--------------------------
'
'This program creates a grid creek network corresponding
'to a given thresold.

'-------------------
'--- Get the View ---
'-------------------
'
theView=av.GetActiveDoc

'---------------------
'--- Get the themes ---
'---------------------

gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
       gridList.add(thm)
     end
 end

fdrthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction grid.",Script.The.GetName)
    if(fdrthm=nil)then
      exit
    end

fdrgrid=fdrthm.getgrid


facthm=Msgbox.ChoiceAsString(gridList,"Choose a flow accumulation grid.",Script.The.GetName)
    if(facthm=nil)then
      exit
    end

facgrid=facthm.getgrid



'Define the creek threshold
thethreshold=msgbox.input("Enter a threshold value to define the creeks (number of cells).",Script.The.GetName,"100")


'Name the creeks grid and coverage

outname1=av.getproject.makefilename("crk_gr","")
aname1=filedialog.put(outname1,"","Creek grid")

outname2=av.getproject.makefilename("crk_cv","shp")
aname2=filedialog.put(outname2,"shp","Creek coverage")


'Define the creeks

streamgrid = (facgrid < thethreshold.asnumber.asgrid).setnull(1.asgrid)
streamgrid.savedataset(aname1)
streamtheme = gtheme.make(streamgrid)
theview.addtheme(streamtheme)
streamtheme.setlegendvisible(false)
```

```
TheFtab=streamgrid.StreamToPolyLineFtab(aname2,fdrgrid,False,prj.MakeNull)
TheFThm=FTheme.Make(TheFtab)
TheView.AddTheme(TheFThm)
TheFThm.setlegendvisible(false)

'-----------
'--- END ---
'-----------
```

'Script: Qual.Creeklimit

```
'
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.Creeklimit
'Version: 1.0
'Date: 11/17/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'---------------------------
'--- Purpose/Description ---
'---------------------------
'
'This program creates a grid representing the limit
'of the creeks associated to a certain threshold.

'--------------------
'--- Get the View ---
'--------------------
'
theView=av.GetActiveDoc

'----------------------
'--- Get the themes ---
'----------------------


gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
       gridList.add(thm)
    end
 end

fdrthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction grid.",Script.The.GetName)
    if(fdrthm=nil)then
      exit
    end

fdrgrid=fdrthm.getgrid


facthm=Msgbox.ChoiceAsString(gridList,"Choose a flow accumulation grid.",Script.The.GetName)
    if(facthm=nil)then
      exit
    end

facgrid=facthm.getgrid


'Define the creek threshold

thethreshold=msgbox.input("Enter a threshold value to define the creeks (number of cells).",Script.The.GetName,"100")

'Name the point grid

outname1=av.getproject.makefilename("crklim","")
aname1=filedialog.put(outname1,"","Creek upstream limit")

'Define the creek limit

streamgrid = (facgrid < thethreshold.asnumber.asgrid).setnull(1.asgrid)
faclimgrid = fdrgrid.flowaccumulation (streamgrid)

'Take the intersection between the cells located in the newly defined
'creeks and the cells whose new flowaccumulation value is 0.

pointlimgrid = (streamgrid = 1.asgrid and  faclimgrid > 0.asgrid).setnull(streamgrid)
pointlimgrid.savedataset(aname1)
pointlimtheme = gtheme.make(pointlimgrid)
theview.addtheme(pointlimtheme)
pointlimtheme.setlegendvisible(false)



'-----------
'--- END ---
'-----------
```

‘ Script Qual.Delete

```
'----------------------------
'--- Creation information ---
'----------------------------
'Name: Qual.Delete
'Version: 1.0
'Date: 10/03/1997
'Author: Christine Dartiguenave
'Center for Research in Water Resources
'The University of Texas at Austin
'darti@crwr.utexas.edu

'---------------------------
'--- Purpose/Description ---
'---------------------------

'Delete several fields in a table

'------------------------
'--- Choose the table ---
'------------------------

doculist = av.GetProject.GetDocs
tablelist=List.Make
      for each d in doculist
        if(d.Is(Table))then
          TableList.add(d)
        end
      end


name=MsgBox.choiceasstring(tablelist, "Choose a table", Script.The.GetName)
if (name = nil)
  then MsgBox.info("No table selected", Script.The.GetName)
  exit
end

'------------------------------------
'---   Choose the fields to delete ---
'------------------------------------

table1=name.getvtab
fieldlist = table1.GetFields
fieldname = MsgBox.MultiListAsString(fieldlist, "Choose the field(s) to delete" ,
Script.The.GetName)
if (fieldname = nil)
  then MsgBox.info("No field(s) selected" , Script.The.GetName)
  exit
end


'-------------------------
'--- Delete the fields ---
'-------------------------

Answer = MsgBox.YesNo("Do you really want to delete these fields?",Script.The.GetName ,
true )
if (answer) then
  table1.seteditable(true)
  table1.removefields(fieldname)
  table1.seteditable(false)
end

'-----------
'--- END ---
'-----------
```

‘Script: Qual.Flow

```
'----------------------------
'--- Creation information ---
'----------------------------

'Name: Qual.Flow
'Version: 1.0
'Creation date: 06/26/97
'Modified 09/16/97
'Modified 10/20/97
'Author: Christine Dartiguenave
'Center for Research in Water Resources
'The University of Texas at Austin
'darti@crwr.utexas.edu


'---------------------------
'--- Purpose/Description ---
'---------------------------

'Compute the base flow, the stormflow and the total flow.in cfs,
'and the grids necessary to the load computation.


'--------------------
'--- Get the view ---
'--------------------
```

```
theView=av.GetActiveDoc
 if (theView.GetThemes.Count = 0) then
    msgbox.error("No themes found", Script.The.GetName)
    exit
  end

'---------------------------
'--- Set analysis extent ---
'---------------------------

'bring up the AnalysisPropertiesDialog
theAE = AnalysisPropertiesDialog.Show(theView,FALSE,"Analysis Properties")
  if (theAE=nil) then
    exit
  end

theExtent = Rect.Make(0@0,1@1)
theCellSize = 1
if ((theAE.GetExtent(theExtent) <> #ANALYSISENV_VALUE) or
    (theAE.GetCellSize(theCellSize) <> #ANALYSISENV_VALUE)) then
  theCE = AnalysisPropertiesDialog.Show(theView,TRUE,"Analysis Extent")
  'check for Cancel from dialog
  if (theCE = NIL) then

    return NIL
  end
  theCE.GetCellSize(theCellSize)
  theCE.GetExtent(theExtent)
end

Grid.SetAnalysisCellSize ( #GRID_ENVTYPE_VALUE , theCellSize )
Grid.SetAnalysisExtent ( #GRID_ENVTYPE_VALUE , theextent )


'-----------------------------
'--- Set working directory ---
'-----------------------------

aProject=av.GetProject
defaultdir=aProject.GetWorkDir
inputdir=MsgBox.Input("Choose the working
directory.",Script.The.GetName,defaultdir.asstring)
  if (inputdir=nil) then
  else
    aDirName = inputdir.asfilename
    aProject.SetWorkDir (aDirName)
  end
```

```
'----------------------
'--- Get the themes ---
'----------------------


'------------------------
'--- Impervious cover ---
'------------------------


icList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if(thm.GetFtab.GetShapeClass.GetClassName="Polygon")then
        icList.add(thm)
      end
    else
      if(thm.is(Gtheme))then
        iclist.add(thm)
      end
     end
  end

icthm=Msgbox.ChoiceAsString(icList,"Choose an impervious cover
theme.",Script.The.GetName)
  if(icthm=nil)then
    exit
  end

'------------------------
'--- Examine IC theme ---
'------------------------

  if (icthm.is(Gtheme)) then
    ic_gr=icthm.getgrid
  else
    theFtab=icthm.getFtab
    theFtab.seteditable(true)
    fieldlist=theftab.getfields
    impc = MsgBox.ChoiceAsString(fieldList,"Choose an IC field.",Script.The.GetName)
      if (impc = nil) then
        Msgbox.info("Can't find IC field in polygon theme",Script.The.GetName)
        exit
     end
  end

  '---------------------------
'--- Flow direction grid ---
'---------------------------
```

```
gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
      gridList.add(thm)
    end
  end

fdrthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction
grid.",Script.The.GetName)

  if(fdrthm=nil)then
    exit
  end

fdirgrid=fdrthm.getgrid

'-----------------------
'--- Correction grid ---
'-----------------------


corrcoefthm=Msgbox.ChoiceAsString(gridList,"Choose a correction
grid.",Script.The.GetName)
  if(corrcoefthm=nil)then
    exit
  end
thethm=corrcoefthm
corrcoef=thethm.getgrid


'----------------
'--- Recharge ---
'----------------

recharge=msgbox.yesno("Do you want to consider a recharge
zone?",Script.The.GetName,true)
  if (recharge=true) then

'Recharge flow

      rechfacthm=Msgbox.ChoiceAsString(gridList,"Choose a recharge grid
(rech_fac).",Script.The.GetName)
        if(rechfacthm=nil)then
          exit
        end
      rechfac=rechfacthm.getgrid


'Recharge cell

      cellrech=Msgbox.ChoiceAsString(gridList,"Choose a cell recharge grid
(lcorr_rech).",Script.The.GetName)
        if(cellrech=nil)then
          exit
        end
      lcorr_rech=cellrech.getgrid

  end

'---------------------------
'--- Precipitation value ---
'---------------------------

'Rainfall amount in in/yr

prec=31.08
preccoef=prec/31.08

'----------------------
'--- Name the grids ---
'----------------------

outname1=av.getproject.makefilename("runoff","")
aname1=filedialog.put(outname1,"","Direct Runoff")

outname2=av.getproject.makefilename("baseflow","")
aname2=filedialog.put(outname2,"","Baseflow")

outname3=av.getproject.makefilename("flow","")
aname3=filedialog.put(outname3,"","Total flow")


'------------------------------------
'--- Calculate runoff coefficients ---
'------------------------------------

  if(icthm.is(ftheme)) then
    rel=msgbox.yesno("Do you want to recompute the runoff coefficients?",
Script.The.GetName, true )
  else
    rel=true
  end

  if (rel) then

'Create the direct runoff coefficient field

      if (icthm.is(ftheme)) then
```

```
      runco = theFtab.findfield("runcoef")
        if (runco = nil) then
          theFtab.seteditable(true)
          runco = field.make("runcoef", #FIELD_DECIMAL, 6, 3)
          theFtab.addfields({runco})
          theFtab.seteditable(false)
        else
          question=Msgbox.yesno("The field runcoef already exists. Do you want to overwrite
it?",Script.The.GetName,true)
            if (question=false) then
              runcofield = Msgbox.input("Name of the direct runoff coefficients
field:",Script.The.GetName,"runcoef1")
                if (runcofield=nil) then
                  exit
                end
              theFtab.seteditable(true)
              runco = field.make(runcofield.asstring, #FIELD_DECIMAL, 6, 3)
              theFtab.addfields({runco})
              theFtab.seteditable(false)
            end
        end


'Create the base flow runoff coefficient field

      runco_bf = theFtab.findfield("runcoef_bf")
        if (runco_bf = nil) then
          theFtab.seteditable(true)
          runco_bf = field.make("runcoef_bf", #FIELD_DECIMAL, 6, 3)
          theFtab.addfields({runco_bf})
          theFtab.seteditable(false)
        else
          question=Msgbox.yesno("The field runcoef_bf already exists. Do you want to
overwrite it?",Script.The.GetName,true)
            if (question=false) then
              runcobffield= Msgbox.input("Name of the base flow runoff coefficients
field:",Script.The.GetName,"runcoef_bf1")
                if (runcobffield=nil) then
                  exit
                end
              theFtab.seteditable(true)
              runco_bf = field.make(runcobffield.asstring, #FIELD_DECIMAL, 6, 3)
              theFtab.addfields({runco})
              theFtab.seteditable(false)
            end
         end

     end

'Calculate runoff coefficient for direct runoff

    rel1 = msgbox.yesno("The ic/runoff coefficient relationship for direct runoff is: runcoef =
0.3428*IC^2 + 0.5677*IC + 0.0125 (0<IC<1) Do you want to change it?" , "Runoff
coefficient " , true)

      if (rel1) then
        labels={"a","b","c"}
        defaults={"0.3428","0.5677","0.0125"}
        coeflist=MsgBox.MultiInput ("runcoef = a*IC^2 + b*IC + c and 0<IC<1", "Direct runoff
coefficients",labels, defaults)
        a1=coeflist.get(0)
        b1=coeflist.get(1)
        c1=coeflist.get(2)
        a=a1.asnumber
        b=b1.asnumber
        c=c1.asnumber


          if (icthm.is(Ftheme)) then
            theFtab.seteditable(true)
            icmax=0
              for each rec in theFTab
                ic1 = theFtab.returnvalue(impc,rec)
                icmax=icmax.max(ic1)
              end
              if (icmax > 1) then
                icperc=true
              else
                icperc=false
              end
              for each rec in theFtab
                ic1 = theFtab.returnvalue(impc, rec)
                  if (icperc=true) then
                    ic1=ic1/100
                  end
                runco1 = (a*ic1*ic1)+(b*ic1)+c
                theFtab.setvalue(runco , rec , runco1 )
              end
            theFtab.seteditable(false)
          else
            aprj=theview.getprojection
            icint=ic_gr.int
            icvtab=icint.getvtab
            icfield=icvtab.findfield("value")
              icmax=0
              for each rec in icvtab
                icvalue=icvtab.returnvalue(icfield,rec)
                icmax=icmax.max(icvalue)
              end
              if (icmax<=1) then
```

```
            runcoef = (a.asgrid*ic_gr*ic_gr)+(b.asgrid*ic_gr)+c.asgrid
          else
            runcoef = (0.0001.asgrid*a.asgrid*ic_gr*ic_gr) + ( 0.01.asgrid * b.asgrid *ic_gr) +
c.asgrid
          end
        end

    else
    'default value
      if (icthm.is(Ftheme)) then
        theFtab.seteditable(true)
          icmax=0
          for each rec in theFTab
            ic1 = theFtab.returnvalue(impc,rec)
            icmax=icmax.max(ic1)
          end
          if (icmax>1)then
            icperc = true
          else
            icperc = false
          end
          for each rec in theFtab
            ic1 = theFtab.returnvalue(impc, rec)
              if (icperc=true) then
                ic1=ic1/100
              end
            runco1 = (0.3428*ic1*ic1)+(0.5677*ic1)+0.0125
            theFtab.setvalue(runco , rec , runco1 )
          end
        theFtab.seteditable(false)

      else
        aprj=theview.getprojection
         icint=ic_gr.int
         icvtab=icint.getvtab
         icfield=icvtab.findfield("value")
         icmax=0
           for each rec in icvtab
             icvalue=icvtab.returnvalue(icfield,rec)
             icmax=icmax.max(icvalue)
           end
             if (icmax<=1) then
               runcoef=(0.3428.asgrid*ic_gr*ic_gr)+(0.5677.asgrid*ic_gr)+0.0125.asgrid
             else

runcoef=(0.0001.asgrid*0.3428.asgrid*ic_gr*ic_gr)+(0.01.asgrid*0.5677.asgrid*ic_gr)+0.012
5.asgrid
          end
      end
    end

'Calculate runoff coefficient for baseflow

rel2 = msgbox.yesno("The ic/runoff coefficient relationship for base flow is: runcoef_bf = -
0.36*IC + 0.1904 if IC < 52% and 0 otherwise (0<IC<1) Do you want to change it?" , "Base
flow runoff coefficients" , true)
  if (rel2) then
    labels={"a","b"}
    defaults={"-0.36","0.1904"}
    bfcoeflist=MsgBox.MultiInput("runcoef_bf = a*IC + b and 0<IC<1","Base flow runoff
coefficients",labels,defaults)
    a1=bfcoeflist.get(0)
    b1=bfcoeflist.get(1)
    a=a1.asnumber
    b=b1.asnumber

      if (icthm.is(Ftheme)) then
        theftab.seteditable(true)
          for each rec in theFtab
            if (icperc=true) then
              ic1=ic1/100
            end
            ic1 = theFtab.returnvalue(impc, rec)
            runco2 = (a*ic1)+b
              if (runco2<0) then
                runco2 = 0
              end
            theFtab.setvalue(runco_bf , rec , runco2 )
          end
        theftab.seteditable(false)
      else
        if (icmax=1 <=1) then
          runcoef1_bf = (a.asgrid*ic_gr)+b.asgrid
        else
           runcoef1_bf = (0.01.asgrid*a.asgrid*ic_gr)+b.asgrid
         end

        runcoef_bf=(runcoef1_bf>0.asgrid).con(runcoef1_bf, 0.asgrid)
      end
  else
    if (icthm.is(ftheme)) then
      theftab.seteditable(true)
        for each rec in theFtab
           ic1 = theFtab.returnvalue(impc, rec)
             if (icperc=true)
               then ic1=ic1/100
             end
           runco2 = -0.36*ic1+0.1904
             if (runco2<0) then
```

```
        runco2 = 0
       end
      theFtab.setvalue(runco_bf , rec , runco2 )
     end
    theftab.seteditable(false)
   else
    if (icmax<=1) then
     runcoef1_bf = (-0.36.asgrid*ic_gr)+0.1904.asgrid
    else
     runcoef1_bf = (-0.0036.asgrid*ic_gr)+0.1904.asgrid
    end
    runcoef_bf=(runcoef1_bf>0.asgrid).con(runcoef1_bf, 0.asgrid)
   end
  end
 end


'-----------------------------------------------------
'--- Create the runoff coefficient grids if needed ---
'-----------------------------------------------------
aPrj=theView.GetProjection
 if (icthm.is(ftheme)) then
  anFtab=icthm.getftab
  aField1 = anFtab.findfield("runcoef")
  runcoef = Grid.MakeFromFTab(anFTab, aPrj, aField1, {thecellSize, theextent})
  aField2 = anFtab.findfield("runcoef_bf")
  runcoef_bf = Grid.MakeFromFTab(anFTab, aPrj, aField2, {thecellSize, theextent})
 end


'--------------------------
'--- Corrected cell flow ---
'--------------------------

'Corrected direct cell runoff

runcoefname = av.getproject.makefilename("runcoef","")
runcoef.savedataset(runcoefname)
runoff = runcoef * corrcoef * 31.08.asgrid * 5.asgrid / 189216.asgrid*preccoef.asgrid
runoffname = av.getproject.makefilename("runcel","")
runoff.savedataset(runoffname)
grid.deletedataset(runcoefname)
runoffgtheme = gtheme.make(runoff)
theview.addtheme(runoffgtheme)
runoffgtheme.setlegendvisible(false)


'Corrected base flow cell runoff (cfs)

if (icthm.is(gtheme)) then
 bfcoef1name = av.getproject.makefilename("bf1coef","")
 runcoef1_bf.savedataset(bfcoef1name)
end

bfcoefname = av.getproject.makefilename("bfcoef","")
runcoef_bf.savedataset(bfcoefname)
baseflow = runcoef_bf * corrcoef * 31.08.asgrid * 5.asgrid / 189216.asgrid * preccoef.asgrid
bflowname = av.getproject.makefilename("bflowcell","")
baseflow.savedataset(bflowname)
grid.deletedataset(bfcoefname)

if (icthm.is(gtheme)) then
 grid.deletedataset(bfcoef1name)
end

bflowgtheme = gtheme.make(baseflow)
theview.addtheme(bflowgtheme)
bflowgtheme.setlegendvisible(false)


'---------------------
'--- Corrected flow ---
'---------------------

'Flowaccumulation

runoff_flow0 = (fdirgrid.flowaccumulation(runoff))
baseflow_gr0 = (fdirgrid.flowaccumulation(baseflow))
totalflow0 = baseflow_gr0 + runoff_flow0

 if (recharge=false) then

  runoff_flow0.savedataset(aname1)
  runoffgtheme = gtheme.make(runoff_flow0)
  theview.addtheme(runoffgtheme)
  runoffgtheme.setlegendvisible(false)

  baseflow_gr0.savedataset(aname2)
  baseflowgtheme = gtheme.make(baseflow_gr0)
  theview.addtheme(baseflowgtheme)
  baseflowgtheme.setlegendvisible(false)

  totalflow0.savedataset(aname3)
  tflow0gtheme = gtheme.make(totalflow0)
  theview.addtheme(tflow0gtheme)
  tflow0gtheme.setlegendvisible(false)


 else
```

```
   runoff0name = av.getproject.makefilename("rnoff0","")
   runoff_flow0.savedataset(runoff0name)

   bflow0name = av.getproject.makefilename("bflow0","")
   baseflow_gr0.savedataset(bflow0name)

   tflow0name = av.getproject.makefilename("tflow0","")
   totalflow0.savedataset(tflow0name)
   tflow0gtheme = gtheme.make(totalflow0)
   theview.addtheme(tflow0gtheme)
   tflow0gtheme.setlegendvisible(false)


'-----------------------------------------
'--- Corrected flow with recharge zone ---
'-----------------------------------------

'Predicted flow

totalflow = totalflow0 - rechfac
totalflow.savedataset(aname3)
totalflowgtheme = gtheme.make(totalflow)
theview.addtheme(totalflowgtheme)
totalflowgtheme.setlegendvisible(false)

'Part of base flow

partbase = baseflow_gr0 / totalflow0
'partbasename = av.getproject.makefilename("partbf","")
'partbase.savedataset(partbasename)
'partbasegtheme = gtheme.make(partbase)
'theview.addtheme(partbasegtheme)
'partbasegtheme.setlegendvisible(false)

'Direct runoff

runcell_rech = lcorr_rech * (1.asgrid - partbase)
runoff_rech = (fdirgrid.flowaccumulation(runcell_rech))
runcrechname = av.getproject.makefilename("rcrech","")
runcell_rech.savedataset(runcrechname)
runoff_flow = runoff_flow0 - runoff_rech
grid.deletedataset(runcrechname)

runoff_flow.savedataset(aname1)
runoffgtheme = gtheme.make(runoff_flow)
theview.addtheme(runoffgtheme)
runoffgtheme.setlegendvisible(false)
grid.deletedataset(runoff0name)

'Base flow

bf_rech = rechfac - runoff_rech
baseflow_gr = baseflow_gr0 - bf_rech
baseflow_gr.savedataset(aname2)
baseflowgtheme = gtheme.make(baseflow_gr)
theview.addtheme(baseflowgtheme)
baseflowgtheme.setlegendvisible(false)
grid.deletedataset(bflow0name)

end

'Message to user

msgbox.info("Flow grids calculated",Script.The.GetName)

'-----------
'--- END ---
'-----------
```

'Script: Qual.HydroZdlsv

```
'*********************************************
' this program assigns to single-cell poly the
' grid-code of its adjcent polygon
'*********************************************
TheProject=av.GetProject
TheView=av.getactiveDoc

ThemeList=TheView.Getthemes
 LineList=list.make
 PolyList=list.make
 for each thm in Themelist
  if(Thm.Is(FTheme))then
   if(thm.getFtab.FindField("Shape").getType=#FIELD_SHAPELINE)then
     LineList.add(thm)
   end
   if(thm.getFtab.FindField("Shape").getType=#FIELD_SHAPEPOLY)then
     Polylist.add(thm)
   end
  end  'if(Thm.Is(Ftheme))then
 end  'for each thm in Themelist
PTheme=msgbox.choiceAsString(Polylist,"Pick a Basin polygon
theme",Polylist.get(0).getFtab.GetName)
  if (Ptheme=nil) then
  exit
  end
LTheme=msgbox.choiceasString(LineList,"Pick a Basin boundary
theme",Linelist.get(0).getFtab.GetName)
  if (Ptheme=nil) then
```

```
 exit
 end

 if (LTheme.Is(FTHEME).not) then
  msgbox.error(LTheme.Getname++"IsNotFtheme","")
  exit
 end
 if (Ptheme.Is(FTHEME).not) then
  msgbox.error(Ptheme.Getname++"IsNotFtheme","")
  exit
 end

PFtab=PTheme.GetFtab
LFtab=LTheme.GetFtab
 if(PFtab.CanEdit)then
  PFtab.SetEditable(true)
 else
  msgbox.info("PFTAB.CAN'TEDIT",Script.The.GetName++"001")
  exit
 end
LLPoly=LFtab.FindField("LPOLY#")
LRPoly=LFtab.FindField("RPOLY#")
if(LLPoly=nil)then
 LLpoly=LFtab.FindField("LPOLY_")
 LRpoly=LFtab.FindField("RPOLY_")
end

PArea=PFtab.FindField("AREA")
PGCODE=PFtab.FindField("Grid-Code")
  if(PGCODE=nil)then
    PGCODE=PFtab.FindField("Grid_Code")
  end
PCovNo=PFtab.FindField(PTheme.GetName+"_")
  if(PCovNo=nil)then
    PCovNo=PFtab.FindField(PTheme.GetName+"#")
   end
if(PcovNo=nil)then
  Pflds=PFtab.GetFields
  PcovNo=msgbox.choiceasString(PFlds,"Select a field as Cov_ or
Cov#",Script.The.GetName)
  if(PcovNo=nil)then
    exit
  end
end
doneDict=dictionary.make(PFtab.GetNumRecords)  'counting for the polys that has been
chked
tmplist=list.make
for each rec in PFtab
 doneDict.add(rec.AsString.AsNumber,0)
 tmplist.add(rec.AsString.AsNumber)
end
' chk=msgbox.choiceAsString(tmplist,"dcnt="+tmplist.count.asString,"OKHERE")
' if (chk=nil)then
'   exit
' end
ddlist=list.make   'holding poly's recnum
kn=0
tnn=PFtab.getNumRecords
for each rec in PFtab

av.ShowMsg(rec.AsString+"of"+tnn.AsString+"looped"+kn.AsString+"RecsAddedToDDlist"
)
 av.SetStatus(rec/(Tnn-1)*100)
  if(doneDict.get(rec)=1)then      'if the record (poly) has been chked, skip
   continue
  end
 GridV=PFtab.ReturnValue(PGCode,rec)
 PFound=False                       'PFound is set to True in the following loop is another
                                    'poly with the same grid-code is found
 tmplist.empty                      'tmplist holds the polys with the same grid-code
 maxarea=0.0
 for each prec in PFtab
   if(doneDict.get(prec)=1)then    'if the Poly has been chked, skip
    continue
   end
   if(Prec=rec)then                'if the Poly is the current Poly, skip
    continue
   end
   GridC=PFtab.ReturnValue(PGCode,prec)   'gets the GCode of the current poly
    if(GridC=GridV)then                   'The current GCode=TheGCode of the poly on the outer
loop
      PFound=true
      if(tmplist.count=0)then             'Put the Poly on the outer loop into the tmplist
        doneDict.set(rec.Asstring.AsNumber,1)
        kn=kn+1
        tmprec=rec.AsString
        maxrec=tmprec.AsNumber
        tmplist.add(tmprec.Asnumber)
        maxarea=PFtab.ReturnValue(Parea,rec)
      end
     tmparea=PFtab.ReturnValue(Parea,prec)
      if(tmparea>maxarea)then
        maxarea=tmparea
        maxrec=prec.Asstring.AsNumber     'Keep track of Max Area Poly
      end
     tmplist.add(prec.AsString.AsNumber)  'Put the current Poly on the inner loop into the list
     doneDict.set(prec.AsString.AsNumber,1)
     kn=kn+1
    end  'endif(gridC=gridV)
 end  'endfor each prec in PFtab
```

```
  if(PFound=true)then                    'Whenever PFound=Ture, keep the poly with MaxArea, the
rest goes to ddlist
    for each ii in tmplist
      if(ii=maxrec)then
       continue
      else
       ddlist.add(ii.AsString.AsNumber)
      end
    end  'end for ii in tmplist
  end 'endif(PFound=true)
end 'for each rec in PFtab
'*******************************************
'assign new grid-code to polygons in the
'dlist
'*******************************************
dcnt=ddlist.count
  chk=msgbox.choiceAsString(ddlist,"dcnt="+dcnt.asString,"OKHERE")
  if (chk=nil)then
   exit
  end
dcrt=0
 if(dcnt=0)then
  msgbox.error("dlistHasZeroMembers",Script.The.GetName++"EXIT-002")
  exit
 end
for each drec in ddlist          'For now,after running this program,one needs to use
ARC/INFO's Dissolve function to dissolve the polys in the ddlinst
 av.setstatus(dcrt/dcnt*100)
 av.showMsg((dcrt+1).Asstring+" of "+dcnt.AsString+" corrected")
 Pcov=PFtab.ReturnValue(PcovNo,drec)
 PFound=False
 nextFound=False
 for each lrec in lFtab
  LpolyV=LFtab.ReturnValue(LLpoly,lrec)
  RpolyV=LFtab.ReturnValue(LRpoly,lrec)
   if(LpolyV=Pcov)then
    PFound=True
    for each prec in PFtab
     PcovC=PFtab.ReturnValue(PCovNo,prec)
      if(PcovC=RpolyV)then
       nextFound=true
       PFtab.SetValue(PGCode,drec,PFtab.ReturnValue(PGCode,prec))
       dcrt=dcrt+1
       break
      end
    end  'for each prec in PFtab
   end   'if(LpolyV=Pcov)then
 if((PFound=True) and (NextFound=true))then
  break
 elseif((PFound=true) and (nextFound=False))then
  chk=msgbox.input("pfound=true,nextFound=False","Pcov="+Pcov.AsString,"EXITLpoly")
  if(chk=nil)then
  exit
  end
  PFound=False
 else
  PFound=False
  NextFound=False
 end
   if(RpolyV=Pcov)then
    PFound=True
    for each prec in PFtab
     PcovC=PFtab.ReturnValue(PCovNo,prec)
      if(PcovC=LpolyV)then
       nextFound=true
       PFtab.SetValue(PGCode,drec,PFtab.ReturnValue(PGCode,prec))
       dcrt=dcrt+1
       break
      end
    end  'for each prec in PFtab
   end   'if(RpolyV=Pcov)then
 if((PFound=True) and (NextFound=true))then
  break
 elseif((PFound=true) and (nextFound=False))then
  chk=msgbox.input("pfound=true,nextFound=False","Pcov"+Pcov.AsString,"EXITRpoly")
  if(chk=nil)then
  exit
  end
  PFound=False
 else
  PFound=False
  NextFound=False
 end
end 'end for each lrec in LFtab
 if(PFound=False)then
  chk=msgbox.input("pfound=False","Pcov="+Pcov.AsString,"EXITforeachLFtab")
  if(chk=nil)then
   exit
  end
 end
end 'end for each drec in ddlist
av.SetStatus(100)
PFtab.SetEditable(false)
''/export/home1/ye/dissolve.822 of 55 lines,
''Written on Wed Sep  6 13:42:55 CDT 1995 by ye
''/export/home1/ye/dissolve.822 of 182 lines,
''Written on Thu Sep  7 15:20:08 CDT 1995 by ye
''/home/ye/sos/dissolve.utl of 182 lines,
''Written on 1996:06:13-09:20:33 by yeZ.
''/home/ye/soflow/dosave/SFdslv.pre of 184 lines,
```

```
''Written on 1992:10:19-10:33:53 by YEZ.


''/appdev/approg/ye/ngflow/dosave/SFdslv.utl of 187 lines,
''Written on 1996:11:05-08:56:24 by YEZ.
```

'Script Qual.Join

```
'----------------------------
'--- Creation information ---
'----------------------------

'Name: Qual.Join
'Version: 1.0
'Date: 10/3/97
'Author: Christine Dartiguenave
'        Center for Research in Water Resources
'        The University of Texas at Austin
'        darti@crwr.utexas.edu

'---------------------------
'--- Purpose/Description ---
'---------------------------

'Join two tables and keep them joined.

'----------------------
'--- Get the tables ---
'----------------------

doculist = av.GetProject.Getdocs
tablelist=List.Make
      for each d in doculist
        if(d.Is(Table))then
          TableList.add(d)
        end
      end


'-------------------------
'--- Destination table ---
'-------------------------

tab1 = Msgbox.ChoiceAsString(tableList,"Destination table",Script.The.GetName)
if (tab1=nil)
  then msgbox.info("No destination table selected", Script.The.GetName)
  exit
end

vtab1 = tab1.GetVTab
fieldlist1 = vtab1.getfields
field1 = MsgBox.choiceAsString(fieldlist1, "Common field for destination table" ,
Script.The.GetName)
if (field1=nil) then
  MsgBox.Info("No field selected", Script.The.GetName)
  exit
end
n=fieldlist1.count

'--------------------
'--- Source table ---
'--------------------

tab2 = Msgbox.ChoiceAsString(TableList,"Source table",Script.The.GetName)
if (tab2=nil) then
  MsgBox.Info("No source table selected", Script.The.GetName)
  exit
end

vtab2 = tab2.GetVtab
fieldlist2 = vtab2.getFields
field2 = MsgBox.choiceAsString(fieldlist2, "Common field for source table" ,
Script.The.GetName)
if (field2=nil) then
  MsgBox.info("No field selected", Script.The.GetName)
  exit
end

'-------------------------
'--- Join the tables ---
'-------------------------

vtab1.Join( field1, vtab2, field2)

'----------------------
'--- Add new fields ---
'----------------------

fieldlist3=vtab1.getfields
p=fieldlist3.count
q=p-1
vtab1.seteditable(true)
i=n

for each i in n.. q
  oldfield=fieldlist3.get(i)
  newfield=oldfield.clone
```

```
  newfieldlist=list.make
  newfieldlist.add(newfield)
  vtab1.addfields(newfieldlist)

    for each rec in vtab1
      val=vtab1.returnvalue(oldfield,rec)
      vtab1.SetValue (newfield, rec, val)
    end

i=i+1
end

vtab1.seteditable(false)


'-----------------------
'--- Remove all joins ---
'-----------------------

vTab1.UnjoinAll

msgbox.info("Tables joined",Script.The.GetName)

'----------
'--- End ---
'----------
```

'Script Qual.load

```
'
'---------------------------
'--- Creation information ---
'---------------------------
'
'Name: Qual.Load
'Version: 1.0
'Date: 05/28/97
'Modified: 10/21/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'--------------------------
'--- Purpose/Description ---
'--------------------------

'This program computes the loads.

'-------------------
'--- Get the View ---
'-------------------

theView=av.GetActiveDoc

if (theView.GetThemes.Count = 0) then
   msgbox.error("No Themes found", Script.The.GetName)
   exit
end


'-------------------------
'--- Set analysis extent ---
'-------------------------

'bring up the AnalysisPropertiesDialog
theAE = AnalysisPropertiesDialog.Show(theView,FALSE,"Analysis Properties")
  if (theAE=nil) then
    exit
  end

theExtent = Rect.Make(0@0,1@1)
theCellSize = 1
if ((theAE.GetExtent(theExtent) <> #ANALYSISENV_VALUE) or
    (theAE.GetCellSize(theCellSize) <> #ANALYSISENV_VALUE)) then
  theCE = AnalysisPropertiesDialog.Show(theView,TRUE,"Analysis Extent")
  'check for Cancel from dialog
  if (theCE = NIL) then

    return NIL
  end
  theCE.GetCellSize(theCellSize)
  theCE.GetExtent(theExtent)
end

Grid.SetAnalysisCellSize ( #GRID_ENVTYPE_VALUE , theCellSize )
Grid.SetAnalysisExtent ( #GRID_ENVTYPE_VALUE , theextent )



'---------------------------
'--- Set working directory ---
'---------------------------
aProject=av.GetProject
defaultdir=aProject.GetWorkDir
```

```
inputdir=MsgBox.Input("Choose the working
directory.",Script.The.GetName,defaultdir.asstring)
 if (inputdir=nil) then
 else
   aDirName = inputdir.asfilename
   aProject.SetWorkDir (aDirName)
 end



'----------------------
'--- Get the tables ---
'----------------------

doculist = av.GetProject.Getdocs
tablelist=List.Make
      for each d in doculist
        if(d.Is(Table))then
          TableList.add(d)
        end
      end


'Direct runoff

emcruntable=Msgbox.ChoiceAsString(tableList,"Choose a direct runoff EMCs
table",Script.The.GetName)
    if(emcruntable=nil)then
      msgbox.error("No direct runoff EMCs table selected",Script.The.GetName)
      exit
    end
emcruntab=emcruntable.getvtab
emcrunlist=emcruntab.getfields


'Base flow

emcbftable=Msgbox.ChoiceAsString(tableList,"Choose a base flow EMCs
table",Script.The.GetName)
    if(emcbftable=nil)then
      msgbox.error("No base flow EMCs table selected",Script.The.GetName)
      exit
    end
emcbftab=emcbftable.getvtab
emcbflist=emcbftab.getfields

'Check that the tables correspond

i=0
  for each rec in emcruntab
    i=i+1
  end
m=i

i=0
  for each rec in emcbftab
    i=i+1
  end
n=i

  if (m=n) then
    p=n-1
    conslist=list.make
    for each i in 0..p
      runcons=emcrunlist.get(0)
      bfcons=emcbflist.get(0)
      runconsname=emcruntab.returnvaluestring(runcons,i)
      bfconsname=emcbftab.returnvaluestring(bfcons,i)
        if (runconsname=bfconsname) then
          i=i+1
          conslist.add(runconsname)
        else
          msgbox.error("The constituents in the two EMCs tables do not
correspond.",Script.The.GetName)
          exit
        end
    end

  else
    msgbox.error("The number of constituents in the two EMC tables is
different.",Script.The.GetName)
    exit
  end


'Choose the constituents to model

choices = MsgBox.MultiListAsString( conslist, "Choose the constituent(s) to model",
Script.The.GetName )
  if (choices = nil) then
    msgbox.info("No constituent selected.", Script.The.GetName)
    exit
  else
    namelist=list.make
    for each cons in choices
        outFName = av.GetProject.MakeFileName(cons, "")
        aName = FileDialog.Put(outFName, "", cons)
          if (aName = Nil) then
            exit
          end
```

```
      namelist.add(aname)
    end
  end



'---------------------
'--- Get the themes ---
'---------------------


undev=0.15

icList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if(thm.GetFtab.GetShapeClass.GetClassName="polygon")then
       icList.add(thm)
      end
    else
      if (thm.is(Gtheme)) then
       iclist.add(thm)
      end

   end
  end



icthm=Msgbox.ChoiceAsString(icList,"Choose an impervious cover
theme.",Script.The.GetName)
    if(icthm=nil)then
      exit
    end

gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
       gridList.add(thm)
     end
  end



fdrthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction
grid.",Script.The.GetName)
    if(fdrthm=nil)then
      exit
    end

fdirgrid=fdrthm.getgrid

zonethm=Msgbox.ChoiceAsString(gridList,"Choose a water landuse theme
(zone_gr).",Script.The.GetName)
    if(zonethm=nil)then
      exit
    end

zone=zonethm.getgrid


runoffthm=Msgbox.ChoiceAsString(gridList,"Choose a corrected runoff cell grid
(runcel1).",Script.The.GetName)
    if(runoffthm=nil)then
      exit
    end

runoff=runoffthm.getgrid


baseflowthm=Msgbox.ChoiceAsString(gridList,"Choose a corrected baseflow cell grid
(bflowc1).",Script.The.GetName)
    if(baseflowthm=nil)then
      exit
    end

baseflow=baseflowthm.getgrid



'Recharge zone

recharge=msgbox.yesno("Do you want to consider a recharge
zone?",Script.The.GetName,true)
  if (recharge=true) then
    lcellrechthm=Msgbox.ChoiceAsString(gridList,"Choose a cell recharge grid
(lcorr_rech).",Script.The.GetName)
      if(lcellrechthm=nil)then
        exit
      end
    lcellrech=lcellrechthm.getgrid

    totalflowthm=Msgbox.ChoiceAsString(gridList,"Choose a total flow grid (without
recharge, tflow01).",Script.The.GetName)
    if(totalflowthm=nil)then
      exit
    end

totalflow=totalflowthm.getgrid
```

```
  end


  if (icthm.is(Gtheme)) then
    ic_gr=icthm.getgrid
  else


'Coverage


'---------------------
'--- Get the table ---
'---------------------


    theFtab=icthm.getFtab
    fieldlist=theftab.getfields
    impc = MsgBox.choiceasstring( fieldlist,"Name of IC field", Script.The.GetName)
      if (impc = nil) then
        addfield1 = msgbox.yesno("Can not find 'IC' field in polygon theme.  Add
it?",Script.The.GetName, true)
        exit
      end
  end


'---------------------------
'--- Calculate EMC values---
'---------------------------

  k=0
  for each cons in choices


    if (icthm.is(Ftheme)) then
      theftab.seteditable(true)

'Check storm runoff field

      consfield = theFtab.findfield(cons+"_[mg/l]")
        if (consfield = nil) then
          consfield = field.make(cons+"_[mg/l]", #FIELD_DECIMAL, 6, 3)
          theFtab.addfields({consfield})
        end

'Check base flow field
```

```
      consfieldbf = theFtab.findfield(cons+"_bf_[mg/l]")
        if (consfieldbf = nil) then
          consfieldbf = field.make(cons+"_bf_[mg/l]", #FIELD_DECIMAL, 6, 3)
          theFtab.addfields({consfieldbf})
        end
    end

'Calculate EMC

'Direct runoff

'Get the parameters a and b (emc=a+b*ic,0<ic<1)
            i=0
            for each rec in emcruntab
              runconsname=emcruntab.returnvaluestring(runcons,rec)
                if (runconsname=cons) then
                  p=i
                else
                  i=i+1
                end
            end

            afield=emcrunlist.get(1)
            bfield=emcrunlist.get(2)
            a=emcruntab.returnvalue(afield,p)
            b=emcruntab.returnvalue(bfield,p)


            if (icthm.is(Ftheme)) then
              icmax=0
              for each rec in theFTab
                ic1 = theFtab.returnvalue(impc,rec)
                icmax=icmax.max(ic1)
              end
            if (icmax>1)then
             icperc = true
           else
             icperc = false
           end

              for each rec in theFtab
                ic1 = theFtab.returnvalue(impc, rec)
                  if (icperc=true) then
                    ic1=ic1/100
                  end

                emcrun=b*ic1+a
                theFtab.setvalue(consfield , rec , emcrun )
              end
            else
```

```
aprj=theview.getprojection
icint=ic_gr.int
icvtab=icint.getvtab
icfield=icvtab.findfield("value")
  icmax=0
  for each rec in icvtab
    icvalue=icvtab.returnvalue(icfield,rec)
    icmax=icmax.max(icvalue)
  end
  if (icmax<=1) then
    emc_gr = ic_gr*b + a.asgrid
  else
    emc_gr = ic_gr*b*0.01 + a.asgrid
  end
end



'Base flow


      afield=emcbflist.get(1)
      bfield=emcbflist.get(2)
      a=emcbftab.returnvalue(afield,p)
      b=emcbftab.returnvalue(bfield,p)

    if (icthm.is(Ftheme)) then
        for each rec in theFtab
          ic1=theftab.returnvalue(impc,rec)
            if (icperc=true) then
              ic1=ic1/100
          end

            if (ic1 <= undev) then
              emcbf=a
            else
              emcbf=b
            end
          theFtab.setvalue(consfieldbf , rec , emcbf )
         end
     else
       if (icmax<=1) then
         emcbf_gr =(ic_gr>undev.asgrid).con(b.asgrid , a.asgrid)
       else
         undev=undev/100
         emcbf_gr = (ic_gr>undev.asgrid).con(b.asgrid,a.asgrid)
       end
     end


'-------------------------
'--- Compute the loads ---
'-------------------------

if(icthm.is(ftheme)) then
  anftab=theftab
  aPrj = theView.GetProjection
end




  if(icthm.is(ftheme)) then

'direct runoff emc

    emc_gr = Grid.MakeFromFTab(anFTab, aPrj, consfield, {thecellSize, theextent})

'baseflow emc

    emcbf_gr = Grid.MakeFromFTab(anFTab, aPrj, consfieldbf, {thecellSize, theextent})

  end


loadcell = runoff *  emc_gr * 3.048.asgrid * 3.048.asgrid * 3.048.asgrid * 86400.asgrid *
365.asgrid / 1000000.asgrid

loadcellbf = baseflow *  emcbf_gr * 3.048.asgrid * 3.048.asgrid * 3.048.asgrid * 86400.asgrid
* 365.asgrid / 1000000.asgrid


'Total
tcellload0 = loadcell+loadcellbf
tcellload=(zone=999).con(0.asgrid,tcellload0)

'Flowaccumulation
load0= (fdirgrid.flowaccumulation(tcellload))

  if (recharge=true) then

    co = load0/totalflow

'Recharge zone
    cellrech = lcellrech * co
    loadrech = (fdirgrid.flowaccumulation(cellrech))

'Total load
```

```
   load = load0-loadrech

  else

   load = load0
  end

aname=namelist.get(k)
load.savedataset(aname)
loadgtheme = gtheme.make(load)
theview.addtheme(loadgtheme)
loadgtheme.setlegendvisible(false)

k=k+1
end

msgbox.info("Load grid(s) calculated",Script.The.GetName)

'-----------
'--- END ---
'-----------
```

'Script Qual.Mergetheme

```
'Name:  View.MergeThemes
'Author: Zichuan Ye
'Title:  Merges two feature themes
'
'Topics:  GeoData

'Description:  Merges the selected themes into a single theme.  A new
'shapefile is created which combines the shapes and attributes of the
'active themes.  The themes to be merged should have the same set of
'attributes (fields).  Only the fields from the first active theme are
'preserved in the output theme.

'Requires:  At least two themes of the same feature type must be in the
'active view.

'Self:

'Returns:

theView   = av.GetActiveDoc
theThemes = theView.GetThemes

if (theThemes.Count < 2) then
  MsgBox.Error( "Must have at least two themes in a view to merge.","")
  exit
end

'Allow the user to choose themes from the view to be merged...
themesToMerge = List.Make
while (true)
  t = MsgBox.Choice( theThemes, "Choose themes in view to merge:"+NL+
    "(Click Cancel to end):", "Merge Themes" )
  if (t <> Nil) then
    themesToMerge.Add(t)
  else
    break
  end
end

if ((themesToMerge = Nil) or (themesToMerge.Count < 2)) then
  MsgBox.Error("Not enough themes to merge.", "")
  exit
end

'Themes must have matching shape types for merging.  Using the first
'active theme verify that this is the case...
checkType  = themesToMerge.Get(0).GetFtab.FindField("Shape").GetType
for each i in 1 .. (themesToMerge.Count - 1)
  t = themesToMerge.Get(i)
  if (checkType <> t.GetFTab.FindField("Shape").GetType) then
    MsgBox.Error("Theme feature type mismatch - Unable to merge.","")
    exit
  end
end

'Specify the output shapefile...
outFName = av.GetProject.MakeFileName("theme", "shp")
outFName = FileDialog.Put(outFName, "*.shp", "Output Merged Shapefile")
if (outFName = Nil) then
  exit
end

  'Create the list of fields used for the output theme.  The fields
'are taken from the first active theme only, it is assumed that
'other themes have an identical set of fields.  If this is not the
'case the themes will still be merged, however fields not found in
'other themes will be empty...

fieldList = List.Make
for each f in themesToMerge.Get(0).GetFTab.GetFields
  if (f.GetName = "Shape") then
    continue
  else
```

```
    fCopy = f.Clone
    fieldList.Add(fCopy)
  end
end

'Get the class of new FTab to create, create the new FTab and
'add fields that we've gathered from the input themes....
shapeType = themesToMerge.Get(0).GetFTab.FindField("Shape").GetType
if (shapeType = #FIELD_SHAPELINE) then
  outClass = POLYLINE
elseif (shapeType = #FIELD_SHAPEMULTIPOINT) then
  outClass = MULTIPOINT
elseif (shapeType = #FIELD_SHAPEPOINT) then
  outClass = POINT
elseif (shapeType = #FIELD_SHAPEPOLY) then
  outClass = POLYGON
else
  MsgBox.Error("Invalid shape field type.", "Merge Themes")
  exit
end
mergeFTab = FTab.MakeNew( outFName, outClass )

if (fieldList.Count > 0) then
  mergeFTab.AddFields( fieldList )
end

'Populate the new FTab from the FTabs of the input themes...
for each t in themesToMerge
  av.ShowMsg( "Merging"++t.GetName )
  inFTab  = t.GetFTab
  if (inFTab.GetSelection.Count = 0) then

    theRecordsToMerge = inFTab
    numRecs = inFTab.GetNumRecords
  else
    theRecordsToMerge = inFTab.GetSelection
    numRecs = theRecordsToMerge.Count
  end
  for each rec in theRecordsToMerge
    av.SetStatus( (rec / numRecs) * 100 )
    newRec   = mergeFTab.AddRecord
    inField  = inFTab.FindField( "Shape" )
    outField = mergeFTab.FindField( "Shape" )
    mergeFTab.SetValue( outField, newrec, inFTab.ReturnValue( inField, rec ))
    if (fieldList.Count > 0) then
      for each f in fieldList
        fName    = f.GetName
        inField  = inFTab.FindField( fName )

        'Skip field if not found in inFTab...
        if ( inField <> Nil ) then
          outField = mergeFTab.FindField( fName )
          aValue    = inFTab.ReturnValue( inField, rec )
          mergeFTab.SetValue( outField, newRec, aValue )
        end
      end   'for each f
    end   'if count
  end   'for each rec
end   'for each t

av.ClearMsg
av.ClearStatus

if (MsgBox.YesNo("Add shapefile as theme to a view?",
  "Merge Themes", true).Not) then
  exit
end

'Create a list of views and allow the user to choose which view to
'add the new theme to...
viewList = {}
for each d in av.GetProject.GetDocs
  if (d.Is(View)) then
    viewList.Add( d )
  end
end

'Include a choice for a new view...
viewList.Add("")
addToView = MsgBox.ListAsString( viewList,"Add Theme to:", "Merge Themes" )

'Get the specified view, make the theme, and add it...
if (addToView <> nil) then
  if (addToView = "") then
    addToView = View.Make
    addToView.GetWin.Open
  end

  mergeTheme = FTheme.Make( mergeFTab )
  addToView.AddTheme( mergeTheme )

  'Bring the View to the front...
  addToView.GetWin.Activate
end
```

‘Script Qual.Pick

```
'
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: pick several.ave
'Version: 1.0
'Date: 08/15/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'---------------------------
'--- Purpose/Description ---
'---------------------------
'
'This program picks up the value in several grids corresponding to a point coverage and write them to the selected
'fields of the attribute table of the coverage.
'
'----------------
'--- Get view ---
'----------------

theView = av.GetActiveDoc

'------------------
'--- Get themes ---
'------------------


if (theView.GetThemes.Count = 0) then
  msgbox.error("No themes found", "Pick")
  exit
end

ThmList=list.Make
for each thm in TheView.GetThemes
  if(thm.is(Gtheme))then
    ThmList.add(thm)
  end
end

Flowthm=Msgbox.multilistasstring(ThmList,"Choose the grid(s)",Script.The.GetName)
    if(Flowthm=nil)then
      exit
    else
      FlowthmName=Flowthm.GetName
    end

Thm1List=list.Make
for each thm in TheView.GetThemes
  if(thm.is(Ftheme))then
   if(thm.GetFtab.GetShapeClass.GetClassName="Point")then
   Thm1List.add(thm)
   end
  end
end

Ptthm=Msgbox.ChoiceAsString(Thm1List,"Choose a point coverage",Script.The.GetName)
    if(Ptthm=nil)then
      exit
     end

pttab = Ptthm.getftab
if (pttab = nil) then
    msgbox.error("Can't open point theme",Script.The.GetName)
    exit

end

'create a field for each grid with the grid name if it does not exist yet.
'If it exists ask to overwrite or to give another name.

for each thm in flowthm
    thmname = thm.getname
    ptvalue = pttab.findfield(thmname)
      if (ptvalue = nil) then
         pttab.seteditable(true)
         ptvalue = field.make(thmname, #FIELD_DECIMAL, 16, 4)
         pttab.addfields({ptvalue})
         pttab.seteditable(false)
       else
         addfield = msgbox.yesno("The field"++thmname.asstring++" already exists. Do you want to overwrite it?",Script.The.GetName, true)
          if (addfield=false) then
            newname = msgbox.input("Enter the new field name", "Field name" , thmname )
            pttab.seteditable(true)
            ptvalue = field.make(newname.asstring, #FIELD_DECIMAL, 16, 4)
            pttab.addfields({ptvalue})
            pttab.seteditable(false)
           end
        end
```

```
  grid1 = thm.getgrid
  ptshape = pttab.findfield("shape")
    if (ptshape = nil) then
      msgbox.error("Can't find 'shape' field in point theme",Script.The.GetName)
    exit
    end

pttab.seteditable(true)
for each rec in pttab
  shapev = pttab.returnvalue(ptshape,rec)
  val = grid1.cellvalue(shapeV,Prj.MakeNull)
  pttab.setvalue(ptvalue,rec,val)
end
pttab.seteditable(false)

end


'
'final message to user
'
message = "Grid values picked"
msgbox.info(message,Script.The.GetName)


'-----------
'--- End ---
'-----------
```

'Script Qual.Wshd

```
'
'----------------------------
'--- Creation information ---
'----------------------------
'
'Name: Qual.Wshd
'Version: 1.0
'Date: 6/26/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'---------------------------
'--- Purpose/Description ---
'---------------------------
'
'This program enable to delineate a watershed based on a point coverage.

'--------------------
'--- Get the View ---
'--------------------
'
theView=av.GetActiveDoc

'----------------------
'--- Get the themes ---
'----------------------


if (theView.GetThemes.Count = 0) then
   msgbox.error("No themes found", Script.The.GetName)
   exit
end

'Name the new grids

outFName = av.GetProject.MakeFileName("wshdgr", "")
aname1 = FileDialog.Put(outFName, "", "Watershed grid")
if (aname1 = Nil) then
  exit
end

outFName = av.GetProject.MakeFileName("wshdcv", "shp")
aname2 = FileDialog.Put(outFName, "*.shp", "Watershed coverage")
if (aname2 = Nil) then
  exit
end



ptList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if(thm.GetFtab.GetShapeClass.GetClassName="Point")then
   ptList.add(thm)
   end
  end
end

ptthm=Msgbox.ChoiceAsString(ptList,"Pick a point coverage",Script.The.GetName)
    if(ptthm=nil)then
      exit
    else
      ptthmName=ptthm.GetName
    end
```

```
gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
       gridList.add(thm)
     end
 end


fdrthm=Msgbox.ChoiceAsString(gridList,"Pick a flow direction grid.",Script.The.GetName)
    if(fdrthm=nil)then
      exit
    end

fdirgrid=fdrthm.getgrid


'-----------------------------------------------------
'--- Convert the point coverage to a grid coverage ---
'-----------------------------------------------------

theftab=ptthm.getftab
cellsize=fdirgrid.getcellsize
box=fdirgrid.getextent
aprj=theview.getprojection

fieldlist=theftab.getfields
afield = Msgbox.ChoiceAsString(fieldList,"Choose a grid-code field.",Script.The.GetName)
    if(afield=nil)then
      exit
    end

ptgrid  = grid.makefromftab(theftab,aprj,afield, {cellsize,box})


'-------------------------------
'--- Delineate the watersheds ---
'-------------------------------

wshdgrid=fdirgrid.Watershed (ptgrid)
wshdgrid.savedataset(aname1)
wshdgtheme = gtheme.make(wshdgrid)
theview.addtheme(wshdgtheme)
wshdgtheme.setlegendvisible(false)

'---------------------------
'--- Convert to shapefile ---
'---------------------------

anFTab = wshdgrid.AsPolygonFtab(aname2,true,prj.makenull)
fthm = FTheme.Make(anFTab)
theView.AddTheme(fthm)
fthm.setlegendvisible(false)

'-----------
'--- END ---
'-----------
```

' Script Qual.Zonalmean

```
'
'---------------------------
'--- Creation information ---
'---------------------------
'
'Name: Qual.Zonalmean
'Version: 1.0
'Date: 10/15/97
'Author: Christine Dartiguenave
'       Center for Research in Water Resources
'       The University of Texas at Austin
'       darti@crwr.utexas.edu
'
'--------------------------
'--- Purpose/Description ---
'--------------------------
'
'This program compute a zonalmean.

'-------------------
'--- Get the View ---
'-------------------
'

theView=av.GetActiveDoc

'
'-----------------
'--- Get themes ---
'-----------------
'


'Check if there are a theme in the view.

if (theView.GetThemes.Count = 0) then
  msgbox.error("No themes found", "BMP")
  exit
```

```
end

outFname = av.GetProject.MakeFileName("grid", "")

gridname = FileDialog.Put(outFName, "", Script.The.GetName)

if (gridname = Nil) then

  exit

end


thmList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
       thmList.add(thm)
    else
      if (thm.is(Ftheme)) then
        if (thm.GetFtab.GetShapeClass.GetClassName="Polygon")then
          thmlist.add(thm)
         end
      end


     end
 end

zoneList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Ftheme))then
      if (thm.GetFtab.GetShapeClass.GetClassName="Polygon")then
        zonelist.add(thm)
       end
    end
  end


  gridList=list.Make
  for each thm in TheView.GetThemes
    if(thm.is(Gtheme))then
           gridlist.add(thm)
       end
     end


zonethm= Msgbox.ChoiceAsString(zoneList,"Choose a zone
coverage.",Script.The.GetName)
  if(zonethm=nil) then
    exit
   end

zonetab=zonethm.getftab
fieldlist=zonetab.getfields
fieldname=Msgbox.ChoiceAsString(fieldList,"Choose a field to identify the
zones.",Script.The.GetName)

zonefield=zonetab.findfield(fieldname.asstring)

valuethm=Msgbox.ChoiceAsString(thmList,"Choose a value theme",Script.The.GetName)
    if(valuethm=nil)then
      exit
    end

if (valuethm.is(Gtheme)) then
  valuegrid=valuethm.getgrid
else
  anftab=valuethm.getftab
  fieldlist1=anftab.getfields
  aprj=theview.getprojection
  fdrthm=Msgbox.ChoiceAsString(gridList,"Choose a flow direction grid (cellsize and
extent)",Script.The.GetName)
  afield=Msgbox.ChoiceAsString(fieldList1,"Choose a value field ",Script.The.GetName)

  fdirgrid=fdrthm.getgrid
  cellsize=fdirgrid.getcellsize
  box=fdirgrid.getextent

  valuegrid = Grid.MakeFromFTab(anFTab, aPrj, aField, {cellSize, box})
end



'-----------------------
'--- Compute the mean ---
'-----------------------

aPrj = theView.GetProjection


meanGrid =
valueGrid.ZonalStats(#GRID_STATYPE_MEAN,zonetab,aPrj,zoneField,FALSE)
meangrid.savedataset(gridname)
meanthm = GTheme.Make(meangrid)

'check for error during operation

if (meanGrid.HasError) then
  return NIL
end
```

```
theView.AddTheme(meanthm)


msgbox.info("New grid computed",Script.The.GetName)

'-----------
'--- END ---
'-----------
```

Glossary

ACV	Annual Capture Volume
AERE	Average Event Removal Efficiency
BMP	Best Management Practices
BOD	Biological Oxygen Demand
COA	City of Austin
COD	Chemical Oxygen Demand
COMP	City of Austin's Composite Ordinance
CRWR	Center for Research in Water Resources
Cu	Copper
CLL	City Limit Line
DEM	Digital Elevation Model
DP	Dissolved Phosphorus
EMC	Event Mean Concentration
EPA	Environmental Protection Agency
EII	Environmental Integrity Index
ETJs	Extra Territorial Jurisdictions
GIS	Geographic Information System
NCDC	National Climatic Data Center
NH_3	Ammonia
NO_3	Nitrate
Pb	Lead
SOS	City of Austin's Save Our Spring Ordinance
TKN	Total Kjeldahl Nitrogen
TN	Total Nitrogen
TNRIS	Texas Natural Resources Information System
TNRCC	Texas Natural Resource Conservation Commission
TOC	Total Organic Compounds
TP	Total Phosphorus
TSS	Total Suspended Solids
TSZ	Traffic Serial Zones
USGS	United States Geological Survey
Zn	Zinc

Bibliography/References

Barrett, M., 1997. *Water Quality and Quantity Inputs for the Urban Creeks Future Needs Assessment*, Draft Report to the City of Austin, CRWR, University of Texas at Austin.

City of Austin Drainage Utility Department, Environmental Resources Management Division, 1997. *Environmental Integrity Index Water Quality Technical Assessment Methodology*, COA-ERM/WRE 1997-03

Chow, V.T., D.R. Maidment, and L. Mays. 1988. Applied Hydrology. McGraw Hill, New York, New York.

Environmental Systems Research Institute, Inc., http://www.esri.com.

Horizons Technology, Inc., http://www.horizons.com.

Maidment, D.R., (ed.), 1993. *Handbook of Hydrology*, McGraw-Hill, New York.

Migalewicz, P.J., and D.R. Maidment, 1996. *Modeling Agrichemical Transport in Midwest Rivers using Geographic Information Systems.* Ph.D. Dissertation, University of Texas at Austin, Department of Civil Engineering, Austin, Texas, CRWR Online Report 96-6 (http://www.ce.utexas.edu/prof/maidment/pub.html).

National Climatic Data Center, http://www.ncdc.noaa.gov.

Olivera, F., D.R. Maidment, and R.J. Charbeneau, 1996. *Spatially Distributed Modeling of Storm Runoff and Non-Point Source Pollution Using Geographic Information System*, Ph.D. Dissertation, University of Texas at Austin, Department of Civil Engineering, Austin, Texas, CRWR Online Report 96-4 (http://www.ce.utexas.edu/prof/maidment/pub.html).

Olivera, F., 1996. Spatial Hydrology of the Urubamba River System in Peru using GIS, http://www.ce.utexas.edu/prof/olivera/peru/peru.htm

Saunders, W.K, and D.R. Maidment, 1996. *A GIS Assesnent of Non-point Source Pollution in the San Antonio-Nueces Coastal Basin.* Masters Report, University of Texas at Austin, Department of Civil Engineering, Austin, Texas , CRWR Online Report 96-1 (http://www.ce.utexas.edu/prof/maidment/pub.html).

Texas Natural Resources Information System, http://www.tnris.state.tx.us/ftparea.html.

USGS, Daily-Value Data for Texas, http://txwww.cr.usgs.gov/cgi-bin/txnwis.

Vita

Christine Dartiguenave was born in Schiltigheim, France on March 14, 1975, the daughter to Michèle Andrée Dartiguenave and Yves Louis Edouard Dartiguenave. After completing her work in 1992 at the Lycée Bellevue, Toulouse, France, she entered the Lycée Pierre de Fermat in Toulouse, France, where she prepared for two years for the national engineering exams, option Mathematics. In 1994, she entered the Ecole Centrale de Lille, Lille, France. During her pursuit of a degree in General Engineering at the Ecole Centrale de Lille, she held an engineer intern position for Thomson-CSF in Toulouse, France in January 1995, and for Elf Aquitaine in Lacq, France, during the 1996 summer. In August 1996, she entered The Graduate School at the University of Texas at Austin. She graduated from the Ecole Centrale de Lille in October 1997, receiving the degree of Ingénieur ECLille.

Permanent address: 15 rue de la Fontaine des Cerdans
31520 Ramonville Saint Agne
France

This thesis was typed by the author.